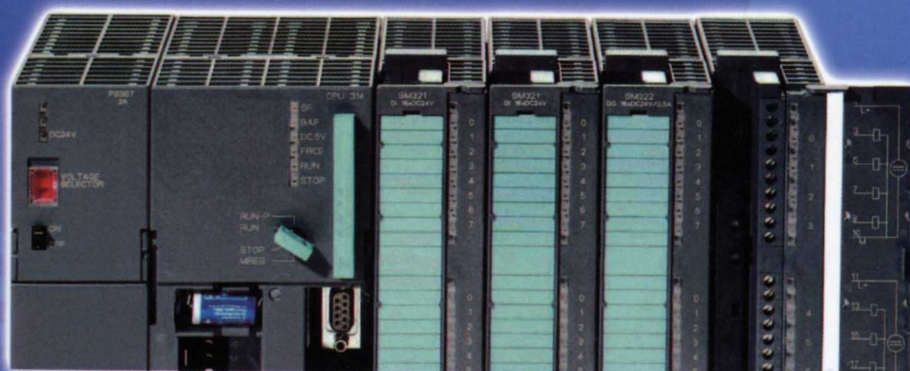
PLC 编程理论·算法及技巧

宋伯生　编著

第2版

PLC BIANCHENG LILUN SUANFA JI JIQIAO







机械工业出版社
CHINA MACHINE PRESS

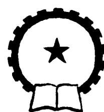
VISIT…

LANZAROTE Caliente.COM

PLC 编程理论、算法及技巧

第 2 版

宋伯生　编著



机 械 工 业 出 版 社

本书较详细地介绍了 PLC 用于顺序控制、过程控制、运动控制、数据处理、联网通信的程序设计理论、方法及技巧，还介绍了 PLC 软件可靠性设计、程序组织及调试等有关问题，既是作者多年从事 PLC 编程经验的全面总结、又是作者深入研究 PLC 编程理论的系统概括。本书列举大量有关编程实例，可直接移植或引用。本书还对 PLC 发展历程、当今面临的挑战与对策也作了深入探讨。

本书第 2 版，针对 PLC 技术的进步，又增加了新内容，并调整了部分结构。在文字上，作者也做了精心修改，更具有可读性。本版还附有光盘。光盘上不仅有本书实例程序、若干上下位机参考程序，还有 OMRON PLC 大量资料及小型机用编程软件。

本书是 PLC 程序设计工程师实用的编程参考用书，也可作高等学校有关专业教师、研究生及本、专科高年级学生的教学参考用书。

图书在版编目（CIP）数据

PLC 编程理论、算法及技巧/宋伯生编著. —2 版—北京：机械工业出版社，2009.3（2011.8 重印）

ISBN 978-7-111-26319-7

Ⅰ.P…　Ⅱ.宋…　Ⅲ.可编程序控制器—程序设计　Ⅳ.TM571.6

中国版本图书馆 CIP 数据核字（2009）第 021620 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）
策划编辑：罗　莉　责任编辑：罗　莉　吕　潇
版式设计：霍永明　责任校对：李　婷
封面设计：姚　毅　责任印制：杨　曦
北京市朝阳展望印刷厂印刷
2011 年 8 月第 2 版第 3 次印刷
184mm×260mm · 43.25 印张 · 1076 千字
5001— 7000 册
标准书号：ISBN 978-7-111-26319-7
ISBN 978-7-89482-999-3（光盘）
定价：88.00 元（含 1CD）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
社服务中心：（010）88361066
销售一部：（010）68326294
销售二部：（010）88379649
读者购书热线：（010）88379203

网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com
封面无防伪标均为盗版

第2版前言

本书发行后，广被学者引用，深受读者厚爱。在中国科技金书网上，本书经该网图书鉴定委员会鉴定，先评定为“经典图书”，后又评定为“优秀教材”。同时也获得网友的广泛赞许。如2006年6月24日，一位网友热评说：“半月前我订购了这本书，它的确是本好书，内容深入浅出，作者实践经验丰富，读者对其内容能较快领会。较一些写得深奥晦涩的实用性书真是好得太多了，我已经看了大半本，受益很多，而且能很容易用在工作中。需要多些这样的科技书。”再如2007年2月9日，有位网友深情地说“这本书我买了。的确很不错，从基础到深入，从理论到实践，从开关量到模拟量，再到脉冲量等等，样样都有，并且讲得也很详细。建议学习工控技术的朋友多看看这本书，对大家PLC技术的提高会有很大的帮助。”对此我深表感谢！

也许是由于本书比较实用，所以正式发行不到几个月，就有读了本书的“热心人”将其“扫描”，制成电子文档，上传到互联网上。很快就有几百人点击下载。后来，本书编辑知道了，通过互联网上聊天工具——“QQ”，与该网站的“斑竹”“友好”协商，该站总算发了“公告”，不再允许下载了。然而，这样“热心人”的后来者还是不断。断断续续，还是不停地出现有不同的“电子版本”，以至于这样的“协商”也难有什么效果。对好心读者、“热心人”这样执着、热情，我也深为感动！

但至今本书出版毕竟已经时过4年。对PLC这样高科技产品，4年的变化是很大的。它的性能又有新提升，功能又有新增强，应用又有新发展，新一代产品又陆续问世！不仅有更多的国外产品进入中国市场，而且我们国产PLC也争得了一席之地。无论是技术层面，还是应用层面，都又有了新进步。显然，作为读者所欢迎的PLC专著，对此应该有所新回应！

这4年，我有幸做了一些PLC应用的理论研究，相继出版了《PLC编程实用指南》、《PLC应用及实验教程》（与陈东旭先生合作）及《PLC系统配置及软件编程》等专著。还在《电气时代》上发表《PLC用于5大控制》、《顺序控制工程方法编程》、《顺序控制同步逻辑法编程》等3篇PLC应用理论文章。同时，也有幸参加一些PLC工程应用的实践，实际编写一些PLC及计算机应用程序。有这样的理论与实践相结合，使我对PLC编程的认知有了不少新的长进。确实，作为一个负责任的作者应该设法也让读者分享这些新收获！

这4年，我还反复阅读本书第1版，并倾听许多读者的关爱意见。有一位著名的电影演员说过，电影艺术是遗憾艺术。影片的不足，只能留下遗憾。不能像话剧那样，上一场的不足，下一场就可补救过来。我感到写书也是如此。在本书第1版中，留下的遗憾，虽在几次重印时有所补救，但最好的补救还是对其修订。

去年秋天，出版社电工电子分社牛新国社长、罗莉编辑考虑到科技书籍的特点，建议我能对本书第1版做些修订。当时，我还没有迫切感，而后又想想，他们是经验之谈，说得真对！既然出版社给了我机会，让我去做一件既能弥补第1版缺憾，又能追踪PLC新进步，也能让读者分享我的新收获的好事，何乐而不为呢！

正是出于上述考虑，今年年初，我开始收集资料，动手修订本书。如今，终于脱稿。主

要在 3 个方面下功夫修改：

(1) 增加新内容，适应 PLC 新进步。增加 PLC 新知识。增加 ST 编程语言及功能块编程知识。增加一些控制新算法及编程实例。增加 C#、VB. Net、Delphi 等可视化语言计算机通信程序。增加电气控制的硬件及 PLC 编程硬件平台知识等。

(2) 调整部分结构，突出算法研究。使全书的系统性有所增强，更便于读者掌握各种编程算法。而编程核心就是设计算法。弄懂了算法，也就有了解决问题的思路、方法及步骤，就不难把实际程序编好。

(3) 精简文字，力争精益求精。同时，还对第 1 版的个别文字及例图错误做了改正，弥补了当时的遗憾。因而，第 2 版内容增加不少，但文字量并不比原来的多。

具体讲，第 1 章增加了继电控制知识。第 2 章增加了 ST 语言及功能块编程介绍，并突出算法概念的叙述。第 3 章更名为顺序控制，增加了组合逻辑编程，结构调整也较大，系统性也得以增强。第 4 章更名为过程控制，增加了 ST 语言及功能块编程。第 5 章更名为运动控制，增加了较多的新算法及新实例。第 7 章通信编程是此次修改内容最多的一章。新内容增加较多，篇幅有较大增加，结构调整也较大。其它各章文字上也都有所修改，文字表达更准确一些。

当然，由于本书只是第 1 版的修订，原有的风格、各章节的原有特点，得到读者肯定的优点是保留的。

一本专著，与其它事物成长一样也要有个过程。“众人拾柴火焰高”，有那么多热心读者的热情呵护和具体帮助，加上我自己的努力，相信会缩短这个过程，会让读者更加满意的！然而，尽管我的决心再大，但个人能力、水平、精力有限，所以也还可能留下遗憾，恳望读者一如既往，不吝赐教！

宋伯生

2008 年 10 月　于沈阳

第 1 版前言

实践经验一再证明，有效地使用 PLC，关键是要处理好两个问题：一是做好系统配置；二是设计好应用程序。系统配置主要是处理好硬件问题；而设计程序则主要是解决好软件问题。

有关系统配置，PLC 的产品说明书多有详细介绍，也可从供应商那里得到相应咨询。所以好办些。但对于设计程序，PLC 的产品说明书至多也只是讲述指令的含义及其使用小例，用它可设计一些简单的程序，而要设计较为复杂的程序就难了。

所以，在用 PLC 建立控制系统时，设计程序，不仅工作量大，而且难度也较大。我在多年的实践中，常遇见这样情况，PLC 硬件系统很快组成了，而它的程序设计没有做好，准备不足，调试时“卡了壳”。这时，如又处于“当事者迷”的状态，一“卡”，可能是好几个星期，甚至不能“自拔”。其不良后果是可想而知的。

事实上，程序设计，或者说编程是一项很细致的工作，对系统的工艺要熟悉，对 PLC 的指令要理解，对 PLC 的资源要清楚。同时，还要弄通有关理论及程序算法。否则无法编好程序，也容易出现“卡壳”这样被动的局面。

本书就是针对这个情况，继作者的《可编程控制器》、《可编程控制器配置・编程・联网》等书之后，为解决这个困惑而编写的又一 PLC 专著。本书以研究程序算法设计为中心，既有设计理论分析、编程方法说明，又有编程技巧推介、实例程序介绍。本书篇幅较大，内容丰富，既是说明 PLC 编程原理的理论书，又是介绍编程技巧的方法书，还是提供众多程序实例的资料书。

本书共分 10 章。

第 1 章讲 PLC 的过去与现在。

第 2 章讲编程基础知识，较详细说明了 PLC 编程指令、操作数、语言、工具、软件及可使用的编程方法，并介绍了 10 种最常用的典型 PLC 梯形图程序。

第 3 章介绍 PLC 开关量控制设计的理论、算法及编程技巧。开关量控制涉及到逻辑设计问题。可使用的处理方法较多，本书几乎都讲到了，并列举了几十个设计实例。相信读者读了这一章后，对弄通逻辑量控制程序设计理论、掌握有关程序设计方法是有所帮助的。

第 4 章介绍 PLC 模拟量控制程序设计的理论及其程序设计方法，并列举大量的设计实例。模拟量控制是自动控制的主题，理论问题很多，实践情况错综复杂，是 PLC 控制程序设计的一个难点。但如果弄通本章介绍的概念和方法，正确地选定控制参数，则有可能设计好相应的控制程序，实现预期的控制要求。

第 5 章介绍 PLC 脉冲量控制程序设计的理论及程序设计方法。脉冲量多在运动系统中使用，所以对它的控制，多与运动控制有关。本章列举大量的这方面设计实例。另外，脉冲量也可用于其它模拟量系统的控制，本章对此也有相应介绍。

第 6 章介绍 PLC 数据处理程序设计的理论、算法及其程序设计方法，列举的设计实例也较多。PLC 用作数据终端是 PLC 的又一应用领域。特别在它实现控制的同时，也兼有数

据终端功能的情况是很多的。所以，设计这个数据处理程序也是 PLC 程序设计的基本功。本章的方方面面论述，将有助于掌握这个基本功。

第 7 章介绍 PLC 与 PLC、计算机、人机界面及智能装置通信的程序设计。联网是当今使用 PLC 的一个趋势，所以通信程序设计也是不可缺少的。为此，本章较详细地列举这些程序的设计要点及通信双方的程序设计实例。

第 8 章介绍 PLC 控制的可靠性设计，篇幅不大，但内容新颖。在当今要求进一步增强 PLC 控制可靠性的新情况下，需要弄通怎样通过编程提高 PLC 控制的可靠性。运用好它，对 PLC 控制的故障避免及其快速排除将有很大帮助。

第 9 章讲 PLC 程序组织与调试。随着 PLC 功能的增强、工作要求的提高，PLC 程序日趋复杂，程序量也不断加大，再加上模块化编程方法的出现，合理组织 PLC 程序是必要的。进一步讲，还有柔性化编程、面向对象编程问题，本章对这些都有所介绍。另外，程序调试、评价也是很重要的，本章对它也有相应说明。

第 10 章讲 PLC 的发展、面对的挑战及应对的策略。

怎么样？从上面列出的内容看，本书确实是涵盖了 PLC 编程的各个方面，可以说在 PLC 运用中所涉及到的编程问题，几乎都讨论到了。

弄通编程理论与编程算法对于 PLC 编程，与学好棋谱对于下棋一样，也是非常重要的。会下棋的人，都有这样的体会，仅了解下棋的规则，而不去研究棋谱，全凭经验去下，是很难有所长进的，甚至往往被讥为“臭棋篓”。同样，如果能多研究一些 PLC 编程，并在理论指导下去实践，那编程的水平将定有长足的进步。

本书对处理各个问题编程的讨论，可以说是编程的“棋谱”，因为它首先都是分析有关理论，然后探讨编程算法，最后才研究代码实现。实践说明，按这个分析理论、设计算法及编写代码的顺序编程，思路清晰，步骤分明，可提高编程水平，并确保编程质量。

本书介绍的编程技巧既是作者理论研究的结果，也是作者多年经验的总结，是很宝贵的。在书中，它的着重点用一段黑体字予以提示。肯定有助于您在编程中少走弯路。

本书在介绍编程理论及算法时，还都列举有程序实例。在本书定稿时，对这些实例多都作了测试。这些实例程序，只要合乎您的情况，可以直接移植，或稍做改动便可引用。

本书篇幅较大，涉及面较广，例子程序又较多，阅读时，请注意弄清概念、抓住要点，多从掌握方法上下功夫。本书每节的开头都标明了该节的关键词，弄清概念请注意弄清这些关键词的含义。

本书所用的实例中产品主要为 OMRON PLC。其实，从编程理论与算法设计的角度讲，什么 PLC 都是一样的。所以，本书提供设计的理论、算法，也可用于其它厂家，如西门子、施耐德、三菱、AB、GE 等公司的 PLC 编程，只要选用好对应的指令、对应的操作数，这些算法的功能也都能实现。

在本书中提到面向对象编程问题。但本书所介绍的编程方法，却都不是面向对象。强调的都是功能、算法，而不是数据、对象，也没有对象的概念。这也是 PLC 当前的编程水平反映。但情况会变化的，PLC 也会用到面向对象编程的，若果真如此，本书只能当作历史资料了，对此我不感懊丧，我还希望这一天早点到来。

这样的事我是体验过的。在 1990 年，我出版《机床控制电路及电控制器》专著后，就遇到过这种情况。该书提出的继电电路设计理论与方法，曾获得较高的评价，但由于继电器-接触器电路逐渐被 PLC 取代，这个设计理论与方法也就逐渐失去实际使用的必要。显然，这个“失去”正是技术进步的表现，不是很好吗？

由于作者水平有限，书中缺点和错误在所难免，恳请广大读者批评指正。

宋伯生

2004 年 8 月于沈阳

目　　录

第 2 版前言

第 1 版前言

第 1 章　可编程序控制器基本知识 ……… 1

1.1　可编程序控制器的产生 ………………… 1

1.1.1　继电控制电路 ……………………… 1

1.1.2　可接插逻辑控制器与顺序控制器 … 5

1.1.3　GM10 条…………………………… 7

1.1.4　PLC 的诞生 ………………………… 7

1.2　可编程序控制器原理 …………………… 8

1.2.1　可编程序控制器实现控制的要点 … 8

1.2.2　可编程序控制器实现控制的过程 … 9

1.2.3　可编程序控制器响应时间计算 …… 13

1.2.4　可编程序控制器实现控制的方式 ………………………………… 14

1.3　可编程序控制器类型 …………………… 14

1.3.1　按控制规模分 ……………………… 15

1.3.2　按结构特点分 ……………………… 15

1.3.3　按生产厂商分 ……………………… 16

1.3.4　按其它特点分 ……………………… 17

1.4　可编程序控制器组成 …………………… 17

1.4.1　CPU 单元 …………………………… 18

1.4.2　内存单元 …………………………… 18

1.4.3　I/O 单元 …………………………… 19

1.4.4　其它单元 …………………………… 20

1.4.5　外部设备 …………………………… 21

1.5　可编程序控制器特点 …………………… 21

1.5.1　功能丰富 …………………………… 22

1.5.2　使用方便 …………………………… 22

1.5.3　工作可靠 …………………………… 23

1.5.4　经济合算 …………………………… 24

1.6　可编程序控制器使用 …………………… 25

1.6.1　系统配置 …………………………… 25

1.6.2　程序编制 …………………………… 34

结语 ………………………………………… 37

请想想 ……………………………………… 37

第 2 章　PLC 编程技术基础 ……………… 38

2.1　PLC 编程语言 …………………………… 38

2.1.1　指令表 ……………………………… 39

2.1.2　梯形图 ……………………………… 39

2.1.3　功能块图 …………………………… 40

2.1.4　连续功能图 ………………………… 41

2.1.5　结构化文本语言 …………………… 41

2.1.6　顺序功能图 ………………………… 42

2.1.7　系统流程语言 ……………………… 43

2.1.8　其它编程语言 ……………………… 45

2.2　PLC 软器件 ……………………………… 46

2.2.1　概述 ………………………………… 46

2.2.2　入出器件 …………………………… 52

2.2.3　内部器件 …………………………… 57

2.3　PLC 指令系统 …………………………… 62

2.3.1　基本逻辑操作指令 ………………… 67

2.3.2　定时、计数指令 …………………… 74

2.3.3　数据处理指令 ……………………… 77

2.3.4　流程控制指令 ……………………… 84

2.3.5　功能块 ……………………………… 92

2.3.6　ST 语言简介 ………………………… 93

2.4　PLC 典型程序 …………………………… 101

2.4.1　起、保、停程序 …………………… 101

2.4.2　状态转换程序 ……………………… 104

2.4.3　定时控制程序 ……………………… 105

2.4.4　动作控制程序 ……………………… 105

2.4.5　步进程序 …………………………… 107

2.4.6　转换程序 …………………………… 109

2.4.7　联锁、互锁程序 …………………… 110

2.5　PLC 编程工具 …………………………… 111

2.5.1　简易编程器 ………………………… 111

2.5.2　图形编程器 ………………………… 117

2.6　PLC 编程软件 …………………………… 117

2.6.1　概述 ………………………………… 117

2.6.2　组成 ………………………………… 118

2.6.3　操作 ………………………………… 124

2.6.4　使用 ………………………………… 125

2.6.5　帮助及其它 ………………………… 137

2.7　PLC 算法编程 …………………………… 137

2.7.1 算法概念 …… 138
2.7.2 算法设计 …… 139
2.7.3 算法表达 …… 139
2.7.4 算法实现 …… 140
2.8 PLC 经验编程 …… 142
2.8.1 经验学习 …… 142
2.8.2 经验积累 …… 143
2.8.3 经验升华 …… 144
2.8.4 经验应用 …… 145
结语 …… 145
请想想 …… 146
请试试 …… 146
第 3 章 PLC 顺序控制程序设计 …… 147
3.1 顺序控制类型及编程方法 …… 147
3.1.1 顺序控制类型 …… 147
3.1.2 顺序控制编程方法 …… 151
3.1.3 顺序控制输入器件 …… 154
3.1.4 顺序控制执行器 …… 155
3.2 组合逻辑设计法编程 …… 158
3.2.1 组合逻辑表达式与真值表 …… 158
3.2.2 组合逻辑分析 …… 163
3.2.3 组合逻辑综合 …… 164
3.2.4 组合逻辑综合实例 …… 165
3.3 异步时序逻辑编程 …… 169
3.3.1 异步时序逻辑表达式与通电表 …… 170
3.3.2 异步时序逻辑分析 …… 173
3.3.3 异步时序逻辑综合 …… 175
3.3.4 异步时序逻辑设计实例 …… 180
3.4 同步时序逻辑编程 …… 190
3.4.1 同步时序逻辑表达式与状态图 …… 190
3.4.2 同步时序逻辑分析 …… 191
3.4.3 同步时序逻辑综合 …… 193
3.4.4 同步时序逻辑状态图法设计实例 …… 194
3.5 图解法编程 …… 203
3.5.1 时序图法编程 …… 203
3.5.2 流程图法编程 …… 207
3.6 高级逻辑设计法编程 …… 213
3.6.1 用字逻辑指令处理 …… 213
3.6.2 用子程序处理 …… 216
3.6.3 用功能块处理 …… 219
3.7 标志逻辑设计法编程 …… 219
3.7.1 基本思路 …… 220
3.7.2 实现方法 …… 220
3.7.3 实际应用 …… 220
3.8 工程设计法编程 …… 223
3.8.1 分散控制及其应用 …… 224
3.8.2 集中控制及其应用 …… 226
3.8.3 混合控制及其应用 …… 230
结语 …… 235
请想想 …… 236
请试试 …… 236
第 4 章 PLC 过程控制编程 …… 237
4.1 过程控制概述 …… 237
4.1.1 PLC 模拟量控制过程 …… 237
4.1.2 PLC 过程控制目的 …… 239
4.1.3 PLC 过程控制类型 …… 239
4.1.4 PLC 过程控制特点 …… 243
4.1.5 PLC 过程控制要求 …… 244
4.2 PLC 模拟量输入及输出 …… 246
4.2.1 模拟量传感器 …… 246
4.2.2 模拟量输入 …… 249
4.2.3 模拟量输出 …… 252
4.2.4 模拟量执行器 …… 255
4.3 开环控制 …… 255
4.3.1 开环特性 …… 255
4.3.2 开环控制 …… 257
4.4 闭环控制 …… 260
4.4.1 ON/OFF 输出控制 …… 261
4.4.2 负反馈控制 …… 262
4.4.3 偏差控制 …… 264
4.4.4 无静差控制 …… 264
4.5 PID 控制 …… 266
4.5.1 PID 控制基本公式 …… 266
4.5.2 PID 控制参数含义 …… 267
4.5.3 PID 控制算法设计 …… 268
4.5.4 PID 控制程序实现 …… 268
4.5.5 PID 控制参数选定 …… 270
4.6 PID 指令及其应用 …… 271
4.6.1 PID 指令说明 …… 272
4.6.2 两个自由度 PID 控制 …… 274
4.6.3 PID 参数选定 …… 274
4.6.4 PID 指令执行 …… 276
4.6.5 PIDAT 指令及其应用 …… 278
4.6.6 使用 PID 指令有关细节 …… 282
4.7 模拟量 PID 硬件单元 …… 285

4.7.1　PID 单元　…… 286
4.7.2　温度控制单元　…… 290
4.7.3　回路控制单元　…… 298
4.8　PID 控制高级应用　…… 303
4.8.1　串级 PID 控制　…… 303
4.8.2　串级比例双辅助回路 PID 控制　… 304
4.8.3　串级比例并交叉限幅双辅回路 PID 控制　…… 305
4.8.4　前馈与 PID 混合控制　…… 305
4.9　模糊控制　…… 306
4.9.1　模糊控制原理　…… 306
4.9.2　模糊控制算法　…… 309
4.9.3　模糊算法实现　…… 312
4.9.4　模糊控制模块　…… 315
4.10　智能控制 …… 322
4.10.1　最优控制 …… 322
4.10.2　自适应控制 …… 324
4.10.3　预测控制 …… 326
4.10.4　学习控制 …… 327
4.10.5　专家控制 …… 330
结语 …… 333
请想想 …… 334
请试试 …… 334
第 5 章　PLC 运动控制编程　…… 335
5.1　概述　…… 335
5.1.1　运动控制的目的　…… 336
5.1.2　运动控制的类型　…… 337
5.1.3　运动控制的特点　…… 338
5.2　脉冲量控制硬件基础　…… 340
5.2.1　脉冲信号生成　…… 340
5.2.2　脉冲信号接收　…… 342
5.2.3　脉冲信号输出　…… 355
5.2.4　脉冲信号执行　…… 369
5.3　闭环控制　…… 374
5.3.1　脉冲量输入开关量输出闭环控制　…… 374
5.3.2　脉冲量输入模拟量输出闭环控制　…… 378
5.3.3　模拟量输入脉冲量输出闭环控制　…… 380
5.3.4　脉冲量输入脉冲量输出闭环控制　…… 381
5.3.5　开关量输入其它量输出闭环控制　…… 382
5.4　开环控制　…… 383
5.4.1　单轴运动控制　…… 384
5.4.2　多轴协调运动控制　…… 387
5.4.3　其它协调运动控制　…… 398
5.4.4　运动控制细节处理　…… 403
5.5　同步控制　…… 404
5.5.1　开环同步控制　…… 404
5.5.2　闭环同步控制　…… 406
5.6　特殊单元控制　…… 407
5.6.1　位置控制单元　…… 407
5.6.2　运动控制单元　…… 423
5.6.3　运动控制器　…… 431
结语 …… 434
请想想 …… 434
请试试 …… 434
第 6 章　PLC 数据处理程序设计　…… 435
6.1　数据终端是 PLC 的新角色　…… 435
6.1.1　专职数据终端实例　…… 435
6.1.2　兼职数据终端实例　…… 438
6.2　数据终端条件及其使用　…… 440
6.2.1　DM 区及对其访问 …… 441
6.2.2　EM 区及对其访问 …… 445
6.2.3　内存卡及对其访问　…… 445
6.2.4　时钟程序　…… 448
6.2.5　时钟设定　…… 449
6.3　数据采集程序设计　…… 451
6.3.1　开关量采集　…… 451
6.3.2　模拟量采集　…… 451
6.3.3　脉冲量采集　…… 453
6.3.4　脉冲选通采集　…… 454
6.3.5　采集数据暂存　…… 456
6.4　数据录入程序设计　…… 457
6.4.1　录入数据设备　…… 457
6.4.2　用通用指令录入　…… 457
6.4.3　用特殊指令录入　…… 458
6.4.4　用编码盘录入　…… 459
6.4.5　用模拟方法录入　…… 460
6.5　数据存储程序设计　…… 461
6.5.1　记录存储　…… 461
6.5.2　压缩存储　…… 464
6.5.3　安全存储　…… 465
6.6　数据显示程序设计　…… 466

6.6.1 数码管数据显示格式 …… 466
6.6.2 数据动态显示 …… 467
6.6.3 简易编程器信息显示 …… 469
6.6.4 数据脉冲选通显示 …… 470
6.6.5 高档数据显示设施 …… 470
6.7 PLC 数据传送 …… 472
6.8 数表处理程序设计 …… 473
6.8.1 求最大、最小数 …… 473
6.8.2 排序 …… 475
6.8.3 求总数 …… 476
6.8.4 求平均数 …… 478
6.8.5 数据查询 …… 478
结语 …… 480
请想想 …… 480
请试试 …… 481
第7章 PLC 通信编程 …… 482
7.1 概述 …… 482
7.1.1 PLC 通信的目的 …… 482
7.1.2 PLC 通信平台 …… 484
7.1.3 PLC 联网通信方法 …… 495
7.1.4 PLC 通信程序特点 …… 499
7.2 PLC 与 PLC 联网通信编程 …… 500
7.2.1 PLC 与 PLC 数据链接通信编程 …… 500
7.2.2 PLC 与 PLC 地址映射通信编程 …… 503
7.2.3 PLC 与 PLC 自由协议通信编程 …… 504
7.2.4 PLC 与 PLC 串行接口协议通信编程 …… 511
7.2.5 PLC 与 PLC 网络协议通信编程 …… 513
7.3 PLC 与计算机通信编程（一） …… 517
7.3.1 串行接口平台通信编程 …… 518
7.3.2 网络平台通信编程 …… 540
7.3.3 互联网通信编程 …… 553
7.3.4 公网平台通信编程 …… 555
7.3.5 工具软件通信编程 …… 556
7.3.6 PLC 方编程 …… 559
7.4 PLC 与计算机通信编程（二） …… 560
7.4.1 组态软件概念 …… 561
7.4.2 组态软件简介 …… 562
7.4.3 组态软件编程 …… 575
7.5 PLC 与人机界面通信编程 …… 582
7.5.1 人机界面概述 …… 582
7.5.2 人机界面方编程 …… 585
7.5.3 PLC 方编程 …… 587
7.6 PLC 与智能装置通信编程 …… 587
7.6.1 用通信指令通信 …… 588
7.6.2 用协议宏通信 …… 589
7.6.3 用从站地址通信 …… 592
结语 …… 592
请想想 …… 592
请试试 …… 592
第8章 PLC 控制可靠性程序设计 …… 593
8.1 概述 …… 593
8.1.1 PLC 控制可靠性概念 …… 593
8.1.2 PLC 控制可靠性类型 …… 594
8.1.3 PLC 控制可靠性意义 …… 595
8.2 PLC 自身工作可靠性 …… 596
8.2.1 PLC 错误类型 …… 596
8.2.2 PLC 系统错误记录 …… 601
8.2.3 PLC 监控指令及其应用 …… 602
8.3 PLC 输入程序可靠性 …… 606
8.4 PLC 输出程序可靠性 …… 610
8.5 PLC 通信程序可靠性 …… 611
8.6 PLC 异常处理程序 …… 612
结语 …… 615
请想想 …… 615
请试试 …… 615
第9章 PLC 程序组织 …… 616
9.1 PLC 程序组织的重要性及方法 …… 616
9.1.1 PLC 程序组织概念 …… 616
9.1.2 PLC 程序组织任务 …… 617
9.2 模块化程序组织 …… 622
9.2.1 程序模块化组织概念 …… 622
9.2.2 使用段、子程序或功能块模块化 …… 622
9.2.3 使用跳转指令模块化 …… 623
9.2.4 使用步进指令模块化 …… 623
9.3 多任务程序组织 …… 623
9.3.1 OMRON PLC 任务划分 …… 623
9.3.2 OMRON PLC 任务管理 …… 624
9.3.3 OMRON PLC 任务组织 …… 626
9.4 PLC 程序柔性化 …… 627
9.4.1 程序使用柔性 …… 627
9.4.2 地址分配柔性 …… 627
9.4.3 参数设定柔性 …… 628
9.4.4 动作选择柔性 …… 629
9.4.5 信号反馈柔性 …… 629

9.5　PLC 面向对象编程 …… 629
9.5.1　面向对象编程概念 …… 630
9.5.2　PLC 面向对象编程设想 …… 631
9.6　PLC 程序调试 …… 634
9.6.1　PLC 程序调试概述 …… 634
9.6.2　PLC 程序仿真调试 …… 635
9.6.3　PLC 程序联机调试 …… 636
9.6.4　PLC 程序现场调试 …… 637
9.6.5　PLC 程序文档 …… 638
9.6.6　PLC 程序保护 …… 638
9.6.7　PLC 程序解密 …… 639
9.6.8　PLC 程序评价 …… 639
结语 …… 642
请想想 …… 642
请试试 …… 643
第 10 章　PLC 在前进 …… 644
10.1　PLC 的性能在提高 …… 644
10.1.1　工作速度在提升 …… 645
10.1.2　控制规模在扩大 …… 646
10.1.3　组成模块在增多 …… 647
10.1.4　内存容量在增大 …… 647
10.1.5　指令系统在增强 …… 648
10.1.6　工作可靠性在提高 …… 648
10.1.7　联网能力在增强 …… 649
10.1.8　外部设备在丰富 …… 650
10.1.9　支持软件在完善 …… 650
10.1.10　经济效益在增加 …… 651
10.2　PLC 的应用在扩展 …… 651
10.2.1　PLC 用于系统控制自动化 …… 652
10.2.2　PLC 用于系统控制远程化 …… 652
10.2.3　PLC 用于系统控制信息化 …… 652
10.2.4　PLC 用于系统控制智能化 …… 653
10.3　PLC 的概念在更新 …… 653
10.3.1　工作模式 …… 654
10.3.2　系统结构 …… 654
10.3.3　设定手段 …… 655
10.3.4　编程方法 …… 656
10.3.5　可靠性设计 …… 657
10.3.6　追求目标 …… 657
10.4　PLC 的类型在增加 …… 659
10.4.1　环境条件扩展型 PLC …… 659
10.4.2　微型 PLC …… 660
10.4.3　分布式 PLC …… 661
10.4.4　内装式 PLC …… 663
10.4.5　安全型 PLC …… 664
10.4.6　运动控制用 PLC …… 665
10.4.7　过程控制用 PLC …… 666
10.4.8　软件 PLC …… 666
10.5　PLC 面临的挑战 …… 667
10.5.1　集散控制系统 …… 667
10.5.2　现场总线控制 …… 669
10.5.3　工业计算机控制 …… 671
10.5.4　其它控制 …… 672
10.6　PLC 去向何处 …… 675
第 1 版后记 …… 679
参考文献 …… 680

第 1 章　可编程序控制器基本知识

可编程序控制器，英文名为 Programmable Controller，简称为 PC，常简称为 PLC，以与个人计算机的英文名 PC（Personal Computer）相区别。本书用 PLC 作为它的简称。本章将对 PLC 的产生、原理、类型、组成、特点、性能、应用及使用进行讨论。

1.1　可编程序控制器的产生

关键词：触点控制电路、继电器控制电路、继电器、接触器、常开触点、常闭触点、可接插逻辑控制器、顺序控制器、GM 八条、可编程序控制器（PLC）

1.1.1　继电控制电路

继电控制电路是指用一些控制电器的触点（如按钮、开关或继电器触点）控制用电器工作的电路。这些控制电器触点及其与用电器的不同连接，可反映它们之间的不同逻辑关系，以实现不同的控制。

继电器控制电路历史悠久，有了电的应用就有它的存在。几经发展，它已越来越便于人们的使用，已在电控领域独领风骚 100 多年的历史了。尽管 PLC 的出现与进步已极大地压缩了它的使用空间。但正如飞机、火车、汽车、自行车的相继出现，前者不能取代后者一样，PLC 的出现，也不可能完全取代继电控制电路。所以，在了解 PLC 的产生的同时，简单地回顾一下继电控制电路是必要的。

控制电器有手动的，它的触点的接通、分断用人工控制，如按钮、手动开关。还有自动的，它的触点的通断能自动实现，如行程开关、继电器。

继电控制电路有触点控制电路及继电器控制电路。以下分别对其进行介绍：

1. 触点控制电路

触点控制电路是指，用手动控制电器触点的接通与分断，去控制用电器的电路。图 1-1 所示的是触点控制电路之一。

图 1-1a 所示为串联电路。它用两个触点串联控制一个电灯 L。如图所示，只有两个按钮同时接通，此灯才能亮。从逻辑的关系讲，灯亮的条件是两个按钮“接通的与”。反之，灯不亮的条件是任意一个按钮分断，即两个按钮“分断的或”。

图 1-1b 所示为并联电路。它用两个触点并联控制一个电灯 L。如图所示，两个按钮任意一个接通，此灯就亮。从逻辑的关系讲，灯亮的条件是两个按钮“接通的或”。反之，灯不亮的条件是两个按钮要同时分断，即“分断的与”。

图 1-1c 所示为混合电路。用触点串联后再并联，去控制一个电灯 L。如图所示，只有 X1、X2 两个按钮同时接通，或按钮 X3、X4 同时接通，此灯可亮。从逻辑的关系讲，灯点亮的条件是“X1、X2 两个按钮接通的与”和“X3、X4 两个按钮接通的与”的“或”。

图 1-1f 所示电路也为混合电路。但它是触点并联后再串联，然后去控制一个电灯 L。如

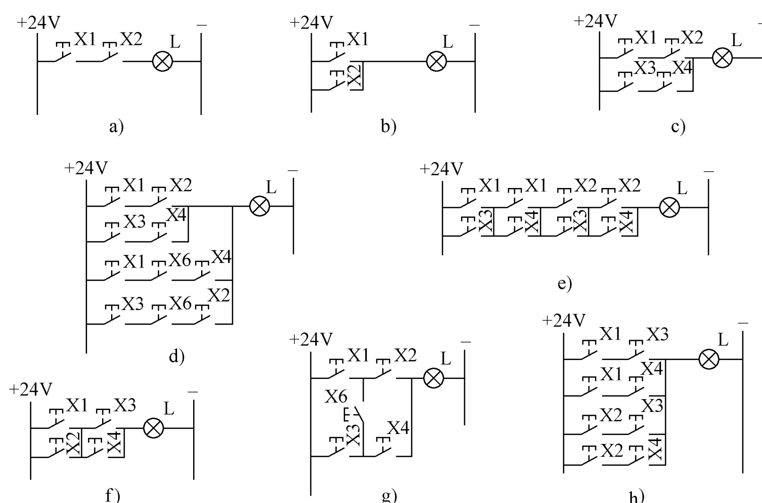


图 1-1 触点控制电路之一

a）串联电路 b）并联电路 c）先串后并电路 d）与 g）桥接等价的先串后并电路 e）与 c）等价的先并后串电路 f）先并后串电路 g）桥接电路 h）与 f）等价的先串后并电路

图所示，只要按钮 X1、按钮 X2 任意一个接通，同时按钮 X3、X4 任意一个接通，则此灯亮。从逻辑的关系讲，灯点亮的条件是“X1、X2 两个按钮接通的或”和“X3、X4 两个按钮接通的或”的“与”。

提示：先串后并到先并后串的转换，可从原电路断的条件确定并的每一个并的小段，然后再把所有小段串联；先并后串到先串后并转换可从原电路的可能通路确定每一个串的支路，然后再把每一支路并联。

图 1-1g 所示为桥接电路。如图所示，除了触点串联、并联还有像按钮 X6 那样的桥接。它不能用简单的“与”、“或”去反映它的逻辑关系，故也称复杂电路。相对于复杂电路，仅用触点串并联的控制电路称为简单电路。

要表达复杂电路的逻辑关系，可根据可能产生的通路情况，先求出在控制功能上，与其等价的简单电路，然后用等价电路的逻辑关系来代表它。图 1-1d 即为图 1-1g 的等价电路。从图知，它除了按钮 X1 与按钮 X2 串联，按钮 X3 与按钮 X4 串联，然后并联之外，还有因按钮 X6 接通，又增加了的两路串联后的并联。

当然也可连接成先并后串的桥接等价电路。这点还是留给读者自己思考吧！

提示：在 PLC 梯形图程序中不允许出现桥接的逻辑关系。

图 1-2 所示为触点控制电路之二。

图 1-2a 所示为先串后并表决电路。这里用了 3 个按钮 X1、X2、X3，当 3 个中任意 2 个，或 3 个全部按下，都将使灯 L 亮。这里只有 3 个按钮参与表决，如果增多，也可实现。只是电路要复杂些。

图 1-2b 所示为先并后串再串比较电路。这里用了两组 4 个按钮 X1、X2 及 Y1、Y2，当这两组接通及分断情况相同时，灯 L 亮。否则，灯 L 不亮。当然，用 3 组，以至于多组比较也是可实现的。只是电路再多串入一组或多组按钮。

图 1-2c 所示为先并后串表决电路，与图 1-2a 的功能相同。

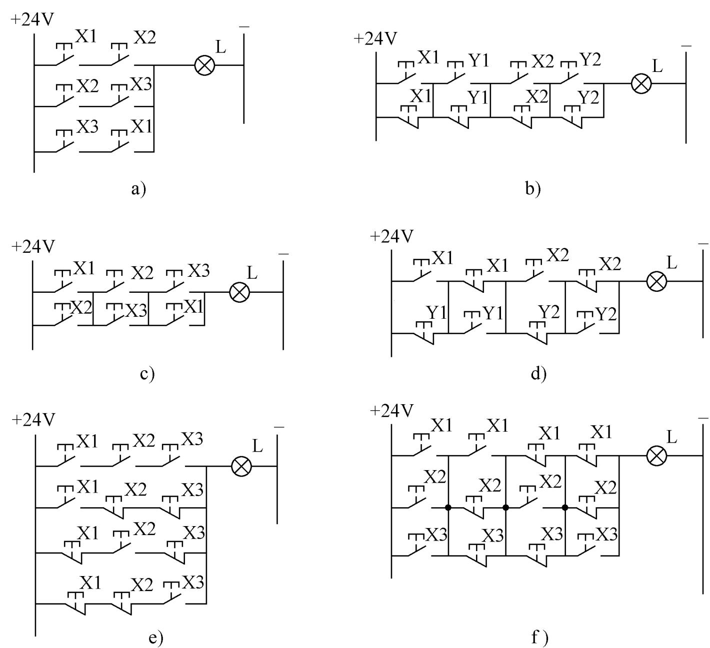


图 1-2　触点控制电路之二

a）先串后并表决电路　b）先并后串再串比较电路　c）先并后串表决电路　d）先并后串比较电路

e）先串后并等权控制电路　f）先并后串等权控制电路

图 1-2d 所示为先并后串比较电路，与图 1-2b 的功能相同。

图 1-2e 所示为先串后并等权控制电路。这里用了 3 个按钮 X1、X2、X3，当 3 个中任意一个改变状态，都将改变灯 L 的状态。把这 3 个开关安装不同位置，可用以实现三地开关对一个灯的等权控制。这里只有 3 个按钮参与等权控制，如果增多，也可实现。只是电路要复杂些。

图 1-2f 所示为先并后串等权控制电路，与图 1-2e 的功能相同。

从以上分析可知，对一些常见的逻辑关系，用触点控制电路是可以实现的。但要实现更复杂的逻辑关系或要用小开关去控制大动力的电路就得求助继电器控制电路了。

2. 继电器控制电路

继电器控制电路除了使用手动控制器触点，还使用继电器触点，其控制对象既有用电器，又有电磁继电器自身线圈。

图 1-3 所示为感应电动机点动控制电路。图 1-3a 为它的工作示意图，图 1-3b 为它的电气原理图。

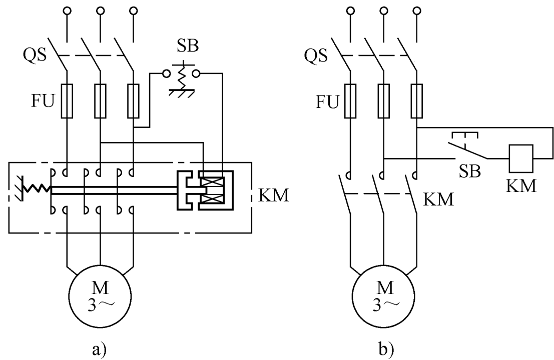


图 1-3　感应电动机点动控制电路

a）示意图　b）电气原理图

从图知，按下按钮 SB，其触点使控制回路接通，接触器 KM 线圈得电。线圈得电，电磁力克服弹簧力，使 KM 主触点接通，电动机 M 得电工作。松开按钮

SB，则控制回路分断，KM 线圈失电，电磁力消失，弹簧力使 KM 主触点分断，电动机 M 失电、停止工作。按钮按下，电动机工作，按钮松开，电动机不工作，所以称之为点动控制。

图中 QS 为刀开关，合上后才能实施控制。FU 为熔断器，用以确保用电安全。

由于这里使用了比按钮触点可通过更大电流的接触器，所以可用一个小小的按钮灵活地控制一个较大功率的电动机。

图 1-4 所示为串联起、保、停电路，是用电钮控制用电器工作最常见的控制电路。

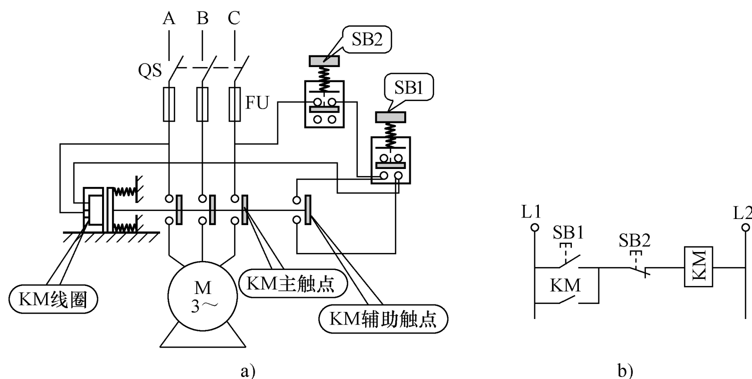


图 1-4 串联起、保、停电路
a）起、保、停电气原理图 b）起、保、停控制电路

从图 1-4a 知，按下按钮 SB1，则控制回路接通，接触器 KM 线圈得电，其常开主触点 KM 接通，电动机 M 得电工作。KM 的常开辅助触点与 SB1 是并联的。因而之后即使松开 SB1，而触点分断，仍可使 KM 线圈得电，继续工作。工作后，按下按钮 SB2，其触点分断，使接触器 KM 线圈失电，进而也使电动机 M 失电而停止工作。由于 KM 的常开辅助触点已分断，之后即使松开 SB2、常闭触点再合上，KM 也仍失电，电动机仍保持停止。可知，此控制电路可控制电动机的起、保、停。

图中，QS 为刀开关，合上才能实现控制。FU 为熔断器，用以确保用电安全。

图 1-4b 所示为图 1-4a 的控制电路。这个电路的要点是，把接触器的常开辅助触点与起动按钮并联，之后再与常闭的停车按钮串联。因此，按钮 SB1 可起动被控设备工作；起动后能使被控设备保持工作；按钮 SB2 可停止被控设备工作。故称之为起、保、停控制电路。

除了上述用串联控制起、保、停电路，还有并联控制及混合控制的类似电路。

图 1-5 所示即为并联起、保、停控制电路。

从图 1-5 可知，这里起动按钮 SB1 改用常闭触点，停车按钮 SB2 改用常开触点。接触器辅助触点也改用常闭触点。并还增加了吸收电阻 R。为了 KM 能正常工作，电源电压要比 KM 的额定工作电压高。

用以上类似的分析，可知，此电路的功能与图 1-4b 完全是相同的。

提示：这里吸收电阻是不可或缺的。不然将出现电源短路，那是绝对不允许的。

图 1-6 所示为混合电路。SB1、SB2、KM 全用其常开触点。它也是用以实现起、保、停逻辑控制。与图 1-5 不同的是，在 KM 停止工作时，吸收电阻不工作，不消耗能量。

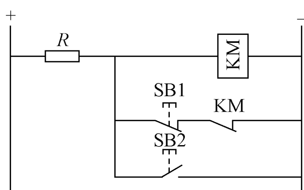


图 1-5　并联起、保、停控制电路

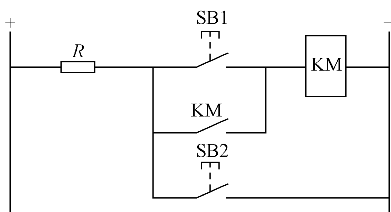


图 1-6　混合起、保、停控制电路

图 1-5 电路进一步演变可得到图 1-7 所示电路。

从图知，其左边画的控制触点 SB1、SB2，为电路输入；控制触点 KM 为接触器 KM 的常闭辅助触点，用作输出反馈。右边画的接触器线圈 KM，为电路输出。中间用点划线框起来的竖线与横线为逻辑方阵。其中 R 为吸收电阻，竖线与横线通过二极管联系（用小斜箭头表示）实现。

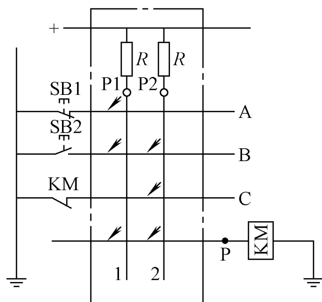


图 1-7　矩阵连接起、保、停电路

如图所示，P 点的高电位是由 P1 与 P2 两点高电位的“或”实现，而 P1、P2 高电位则取决于控制它的触点分断的“与”。可以看出，如图所示的逻辑关系，也是起、保、停逻辑。

图 1-7 所示的电路用的元器件多，但可把控制电路“标准化”了。从设计思想上讲，这里有两个突破：

（1）便于更改控制逻辑关系；

（2）控制触点使用次数可不受限制。

这两条突破，可使控制电路实现各种逻辑控制变得方便、灵活，是很有意义的。

使控制电路便于设计，便于更改，有人还曾作过其它方面的探讨。有的还设计成可进行程序预选及反馈预选的电路。然而，它总是要靠触点的通断实现控制，所以不可避免地存在如下一些缺点：

（1）触点转换总是需要时间的，总是有电滞后及机械滞后。这两个滞后加起来，长的可达几十毫秒，甚至更多。这两个滞后可能造成控制不及时、不同步。不及时，会降低控制精确度；不同步，有时会使电路工作出现竞态。这是在电路设计时，不得不采取措施加以避免的。

（2）触点频繁通断，易受电火花烧蚀与机械磨损，并因此随着使用次数的增加，总是会带来一些故障。

（3）体积大，安装这些元器件有时需庞大的控制柜。

（4）消耗电能多。初次投资费不大，但日常使用维护费很高。

（5）难以或不便实现逻辑关系复杂的控制。

（6）电路不通用，每种电路多数都要单独设计、单独制造，更改也不便。

所以，要实现复杂的、灵活的、更可靠的、小型化的电气控制，必须寻找别的出路。

1.1.2　可接插逻辑控制器与顺序控制器

1. 可接插逻辑控制

在图 1-7 电路中，吸收电阻、隔离二极管是直接与继电器线圈串联的。继电器工作时，

通过它们的电流都相等；继电器不工作时，通过吸收电阻的电流则更大。这既浪费电能，又得考虑散热问题，所以不大适用。

图 1-8 电路是它的改进。图 1-8a 的继电器线圈通过晶体管带动。吸收电阻、隔离二极管与带动继电器工作的晶体管的基极相连，所以消耗的功率小得多。

图 1-8b 用 RS 触发器代替继电器。输出为无触点元件。不仅消耗功率低，而且转换速度快，工作可靠，工作寿命也长。

在这个基础上，还可用一些集成电子元件，使控制器小型化，成为可编程序逻辑控制器，可方便地用以实现输出与输入之间的逻辑变换。但是，它的变换编程只能通过逻辑方阵中竖线与横线的不同连接实现，所以实际上应称之为可接插逻辑控制器。

2. 可编程序顺序控制器

搞过逻辑设计的人都有这样体会，用逻辑运算的方法做逻辑综合是比较难的。特别是对时序逻辑，处理起来更难。而用程序步进控制的思路处理它，则比较容易。

图 1-9 所示为程序步进控制电路。图中的程序步进器按顺序 1、2、3…转换程序。程序与输出的对应关系通过逻辑方阵（图左下方）预选。如图所示，程序 1 与输出 B 相连、程序 2 与输出 C 相连，即执行程序 1 时，可使输出 B 工作，等等。要改变程序与输出的关系，只要改变这里的竖线与横线间连接就可以了。

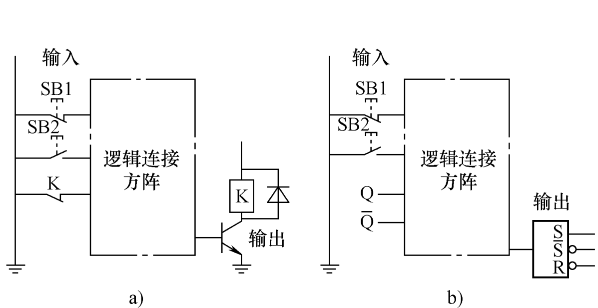


图 1-8 可接插控制逻辑

a）晶体管驱动输出 b）RS 触发器驱动输出

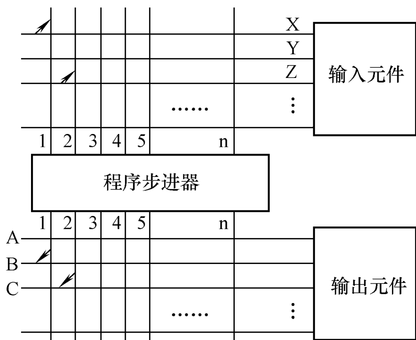


图 1-9 程序步进控制电路

程序步进器的步进，或叫程序转换，由输出动作完成的应答信号，即输入信号控制。而执行某个程序后到底由哪个应答信号控制步进，也通过逻辑方阵（图左上方）预选。如图所示，执行程序 1 时，输入 X 动作可使程序步进；执行程序 2 时，Z 可使程序步进，等等。要改变这个关系，用改变横线与竖线的连接即可实现。

实现图 1-9 关系的具体电路很多，而且都用了集成电子电路，由于它是按程序顺序工作的，程序的动作又是可更改的，故称之为顺序控制器。

上述这两种控制器的出现，把控制电路发展成了电子控制器。它可由专门厂家成批生产，用户购置后，依需要设计好逻辑关系，稍作接线即可使用。这应该说是电气控制的一大进步。

不过，它们编程是靠硬件实现的，还是不大方便。同时，大规模集成电路也难以使用。这使得这种控制器的进一步小型化、降低成本、降低功耗，以及获得更高的可靠性都受到限制。也正是由于这些原因，这两种控制器虽然也风行过一时，但最后还是被淘汰了。

1.1.3　GM10 条

20 世纪 60 年代末期，由于电子计算机技术的发展，曾把小型电子计算机用作机械工作或生产过程的逻辑控制。计算机通过改变程序，更改控制，这比更改硬件接线方便得多。但它也存在一些缺陷：

（1）编程复杂，要求有较高水平的编程与操作人员；

（2）须要配备相应的外部接口；

（3）对工作环境的条件要求较高；

（4）功能“过剩”，机器资源未能充分利用；

（5）造价昂贵。

所以，美国通用汽车（GM）公司为适应汽车工业发展的需要，于 1968 年提出设计新型电控制器的要求，并提出 10 点招标指标：

（1）编程简单，可在现场修改程序；

（2）维护方便，最好是插件式；

（3）可靠性高于继电器控制柜；

（4）体积小于继电器控制柜；

（5）可将数据直接送入管理计算机；

（6）在成本上可与继电器控制器竞争；

（7）输入可以是交流 115V；

（8）输出为交流 115V、2A 以上，能直接驱动电磁阀；

（9）在扩展时，原有系统只需很小变更；

（10）用户程序存储器容量至少能扩展到 4k 字。

这就是著名的 GM10 条。如果说种种电控制器、电子计算机技术的发展是 PLC 出现的物质基础，那么 GM10 条则是 PLC 出现的直接原因。

1.1.4　PLC 的诞生

GM10 条提出后，美国数据设备公司（DEC）用一年多时间的努力，研制出第一台这种电控制器。它在美国 GM 公司的装配线上试用，取得了成功。

从此，这项技术就迅速地发展起来。过了两年，日本人从美国引进了这项新技术，很快研制成类似的控制器，即 DSC-8。又过了两年，即 1973 年，西欧国家也制成这样的控制器。

有决定意义的进步还是在 1975 ~ 1976 年之间。这两年，美国、日本、西德等一些国家把微处理器（Microprocessor）用作可编程序控制器的中央处理单元（Center Processing Unit，CPU），用集成电路的存储器代替磁心存储器，把微型计算机技术与上述两种电控制器结合起来，使得可编程序控制器实现更大规模的集成化，工作更为可靠，更能适应工业环境，而且还更加灵活、功能也更加强，同时成本也大幅度地降了下来，从而使 PLC 进入了实用阶段。

我国于 1974 年开始研制 PLC，1977 年开始工业应用。

这种控制器出现后，其名称和定义不大统一。为此，美国电气制造商协会（National Electrical Manufacturers Association，NEMA）于 1980 年正式命名其为可编程序控制器

(Programmable Controller)，简称 PC，并给它下了定义：“PC，是一种数字式电子装置，它使用了可编程序的记忆体以存储指令，用以执行诸如逻辑、顺序、定时、计数与演算等功能，并通过数字或类似的输入/输出模块，以控制各种机械或工作程序。”

1985 年 1 月，国际电工委员会（IEC）在颁布可编程序控制器标准草案第 2 稿时，对它也下了定义。这个定义为：“可编程序控制器是一种数字运算操作的电子系统，专为在工业环境下应用而设计。它采用可编程序的存储器，用来在其内部执行逻辑运算、顺序控制、定时、计数和算术运算等操作指令，并通过数字式、模拟式的输入和输出，控制各种类型的机械或生产过程。”

可知，PLC 这个电子系统，也是靠存储程序、执行指令，进行信息处理，实现输入到输出的变换。但它的目的是用以控制各种类型机械或生产过程。所以，从实质上讲，它是一台工业环境应用的、满足实时控制要求的专用计算机。与普通计算机所不同的主要是：

它没有键盘，代之为一个个输入电路，并用其获取控制命令或现场信号。同时，此输入电路具有滤波能力，与内部电路为电隔离，但可通过光耦合建立联系；

它没有显示器，代之为一个个输出电路，并用其产生控制输出。由于此电路具有驱动能力，故可以驱动一般的工业控制元器件，如电磁阀、接触器等。同时，此电路与内部电路也是电隔离的，用光或磁耦合建立联系；

它没有硬盘，只有内存。但可配备存储卡，以为程序与数据建立备份；

它配置有外设或通信接口，可用以编程或下载程序、监控及联网通信；

它的结构为模块化，体积小，安装方便，比较坚固，具有很强的抗干扰、抗冲击、抗振动特性。

总之，PLC 只是一台没有键盘、没有显示器、没有硬盘，但有很多输入、输出电路，配有接口，可在工业现场实时使用的、模块化、小型化的特殊计算机。

要指出的是，随着技术进步，PLC 的功能在不断增强，性能在不断提高，应用在不断扩展，类型在不断增多。所以，它的概念也在不断更新。无疑的是，它已发展成为当今方方面面自动化、信息化的重要支柱。

1.2　可编程序控制器原理

关键词：基于电子计算机、输入输出信息变换、可靠物理实现、输入暂存器、输入继电器、输出锁存器、输出继电器、输入刷新、输出刷新、扫描方式、立即刷新、中断方式

1.2.1　可编程序控制器实现控制的要点

它的工作有两个要点：输入输出信息变换、可靠物理实现。

输入输出信息变换主要由运行存储于 PLC 内存中的程序实现。这类程序既有系统的（这类程序又称监控程序或操作系统），又有用户的。系统程序为用户程序提供编辑与运行平台，同时，还进行必要的公共处理，如自检，I/O 刷新，与外设、上位计算机或其它 PLC 通信等处理。用户程序由用户按照控制的要求进行设计。什么样的控制，就有什么样的用户程序。可知，输入输出信息变换主要是软件问题。

可靠物理实现主要通过输入（I，Input）及输出（O，Output）电路。每一输入点或输出

点就有一个 I 或 O 电路。而且，总是把若干个这样电路集成在一个模块（或箱体）中，然后再由若干个模块（或箱体）集成为 PLC 完整的 I/O 系统（电路）。尽管这些模块相当多，占了 PLC 体积的大部分，但由于它们都是高度集成化的，所以 PLC 的体积还是不太大的。可知，可靠物理实现主要是硬件问题。

输入电路时刻监视着输入点的通或断状态，并将此状态暂存于它的输入暂存器（还可能有别的称谓）中。每一输入点都有一个与其对应的输入暂存器。

输出电路有输出锁存器（还可能有别的称谓）。它有高、低电位两个状态，并可锁存。同时，它还有相应的物理电路，可把这个高、低电位状态传送给输出点。每一输出点都有一个与其对应的输出锁存器。

这里的暂存器及锁存器实际是 PLC 的 I/O 电路的寄存器。它们与 PLC 内存交换信息是通过 PLC I/O 总线及运行 PLC 的系统程序实现的。

把暂存器的信息读到 PLC 的内存中，称为输入刷新。PLC 内存有专门开辟的存放输入信息的映射区。这个区的每一对应位（bit）称为输入继电器，或称软触点，或称为过程映射输入寄存器（Process-image Input Register）。这些位（bit）置成 1，表示触点通，置成 0 表示触点断。它的状态是由输入刷新得到的，所以它反映的就是输入点的状态。

锁存器与 PLC 内存中的输出映射区是对应的。一个锁存器有一个与其对应内存位（bit），这个位称为输出继电器，或称输出线圈，或称为过程映射输出寄存器（Process-image Output Register）。通过 PLC I/O 总线及运行系统程序，输出继电器的状态将映射给锁存器。这个映射的完成也称为输出刷新。

PLC 除了有可接收开关信号的输入电路，有时还有接收模拟信号的输入电路（称为模拟量输入单元或模块）。只是后者先要进行模/数转换，然后，再把转换后的数据存入相应的 PLC 内存单元中。

如要产生模拟量输出，则要配有模拟量输出电路（称为模拟量输出模块或单元）。靠它对 PLC 相应的内存单元的内容进行数/模转换，并产生输出。

这样，用户的程序只是，PLC 输入内存区到输出内存区的变换。这是一个数据及逻辑处理问题。由于 PLC 有强大的指令系统，编写出满足这个要求的程序是完全可能的。

图 1-10 对以上叙述作了说明。其中框图代表信息存储的地点，箭头代表信息的流向及实现信息流动的手段。这个图既反映了 PLC 实现控制的两个基本要点，同时也反映了信息在 PLC 中的空间关系。

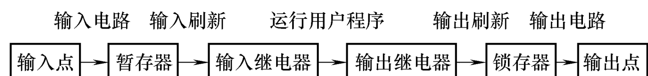


图 1-10　PLC 实现控制示意图

1.2.2　可编程序控制器实现控制的过程

简单地说，PLC 工作过程是：输入刷新→运行用户程序→输出刷新，再输入刷新→再运行用户程序→再输出刷新……永不停止循环反复地进行着。

图 1-11a 所示的流程图反映的就是上述过程。它也反映了信息间的时间关系。

有了上述过程，用 PLC 实现控制显然是可能的。因为有了输入刷新，可把输入电路得到的输入信息存入 PLC 的输入继电器；经运行用户程序，输出继电器将得到变换后的信息；经输出刷新，输出锁存器将反映输出映射区的状态，再通过输出电路将产生输出。又由于这

个过程是循环反复地进行着，所以输出总是会响应输入的变化。只是响应的时间上略有滞后。

提示：工作速度快、执行指令时间短，是 PLC 实现控制的基础。没有工作的高速度也就没有 PLC。

事实上，它的速度是很快的，执行一条指令，长的几微秒、几十微秒，短的才零点几，或零点零几微秒。而且这个速度还在不断提高中。

图 1-11a 所示的是简化的过程，实际的 PLC 工作过程还要复杂些。除了 I/O 刷新及运行用户程序，还要做些其它的公共处理。如：循环时间监视、外设服务及通信处理等。

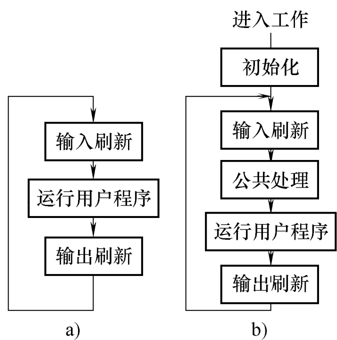


图 1-11　PLC 工作流程图
a）简化工作流程图　b）实际工作流程图

（1）循环时间监视。是用“看门狗”（Watching Dog），这也是一般微机系统常用的做法。具体的是设一个定时器，监测用户程序的运行时间，只要循环超时，即报警或作相应处理。

（2）外设服务。是让 PLC 接受编程器对它的操作，或通过接口向输出设备（如打印机）输出数据。

（3）通信处理。实现与计算机，或与其它 PLC，或与智能操作器、传感器进行信息交换，这也是增强 PLC 控制能力的需要。

总之，PLC 工作过程总是：公共处理→I/O 刷新→运行用户程序→再公共处理→……反复不停地重复着。

此外，如同普通计算机，PLC 上电后，也要进行系统自检及内存的初始化工作，以保证 PLC 正常工作。

把上述过程综合起来，较完整的 PLC 工作过程大体如图 1-11b 所示。

到此，不妨用图 1-12 所示的，按钮 SB1、SB2 通过 PLC 控制接触器 KM 工作例，看看 PLC 的这个工作过程。本例用的 PLC 为 OMRON CPM2A 小型 PLC。

1. 接线

图 1-13 所示为输入输出接线图。该图上方为输入通道（00CH）及其输入点，下方为输出通道（10CH）及其输出点。本例只是用以说明 PLC 的工作过程，故仅使用 0.00 、0.01 两个输入点及 10.00 一个输出点。

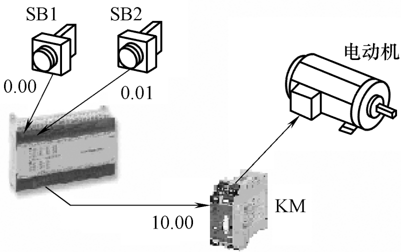


图 1-12　用 PLC 实现控制过程示意图

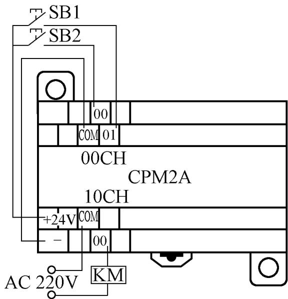


图 1-13　输入输出接线图

0.00 为 00CH（0 通道）的 00 点，0.01 为 00CH（0 通道）的 01 点。如图 1-12 所示，起动按钮 SB1 的一端与 0.00 连接，停止按钮 SB2 的一端与 0.01 连接。两个按钮另一端都与

直流电源 +24V 端连接。直流电源的负端则与 COM（公用）端相连。

提示：电器触点与 PLC 的输入点相接，一般都用常开触点。

10.00 为 10 CH（10 通道）的 00 点。如图 1-13 所示，接触器 KM 线圈的一端与它连接，线圈的另一端与 220V 电源电压连接。而电源的另一端则与 COM（公用）端相连。

输入点到 COM 端的电路为输入电路。输出点到 COM 端的电路为输出电路，该图中都没有画出。

有了这样输入接线，只要按钮闭合，其相应的输入电路将产生约 8～10mA 的电流，经输入刷新，可使对应的输入继电器置 1。当然，如果按钮未按下，回路不通，没有输入电流，对应的输入继电器将置 0。

有了这样输出接线，输出继电器为 1，经输出刷新，这个电路接通，接触器 KM 将得电、工作。输出继电器置为 0，经输出刷新，这个电路将分断，接触器 KM 将失电、停止工作。

2. 程序

图 1-14 所示为实现上述控制的梯形图程序。它很像继电控制电路，有触点，也有线圈。其关系与上述起、保、停逻辑很相似。而表 1-1 为与其对应的助记符程序。本例只是用以说明 PLC 的工作过程，故该程序仅含两条 5 个指令，见表 1-1。

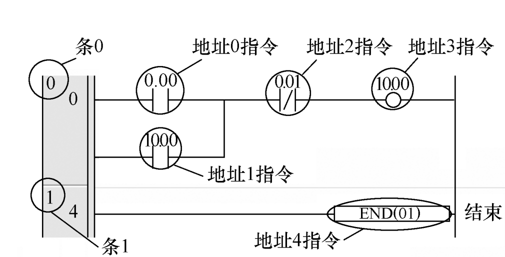


图 1-14　梯形图程序

表 1-1　助记符指令

条	地址	指令码	操作数
0	0	LD	0.00
	1	OR	10.00
	2	AND-NOT	0.01
	3	OUT	10.00
1	4	END	

地址 0 指令，是装载指令。含义是把输入继电器 0.00 的内容（1 或 0）存入结果寄存器 R 中。

地址 1 指令，是或指令。含义是把结果寄存器 R 的内容与输出继电器 10.00 的内容作“或”运算，然后再把结果存入结果寄存器 R 中。

地址 2 指令，是非与指令。含义是对输入继电器 0.01 内容取非（反）后，再与 R 的内容作“与”运算，然后再把结果存于 R 中。

地址 3 指令，是输出指令。含义是把结果寄存器 R 的内容赋值给输出继电器 10.00。

地址 4 指令，是结束指令。含义是程序执行到此为止，一个扫描周期的用户程序运行结束。

3. 运行

把上述程序送入 PLC，令其运行，用按钮 SB1、SB2 即可控制接触器了。以下分 4 种情况进行分析：

（1）情况 1。若 SB1 按钮合上，SB2 按钮没合上。

经输入刷新，0.00 输入继电器置 1，输入继电器 0.01 置 0。这时：

执行地址 0 指令，把 0.00 的内容装载到结果寄存器 R 中，结果寄存器 R 的内容为 1。

执行地址 1 指令，这时 R 的内容已为 1，故不管 10.00 内容是什么，执行它后，R 的内容还为 1。

执行地址 2 指令，对操作数取“非”后，再作与的运算。0.01 的内容为 0，取“非”后为 1。1 与 1 相与仍为 1。故，执行它后，R 的内容还为 1。

执行地址 3 指令，把结果寄存器 R 为 1 的值赋值给输出继电器 10.00，故执行它后，输出继电器的内容为 1。

执行地址 4 指令，用户程序结束，开始输出刷新。10.00 为 1，输出点与 COM 接通，使 KM 得电而工作。

显然，只要情况 1 继续保持，再经历一个新的周期时，只是 10.00 变为 1。它改变的只是地址 1 指令的操作数。而这个操作数的改变，并不改变原来的结果。

（2）情况 2。若在经历情况 1 后，按钮 SB1 松开，按钮 SB2 也不合上。

经输入刷新，输入继电器 0.00 置 0，0.01 输入继电器置 0。这时：

执行地址 0 指令，把 0.00 的内容装载到结果寄存器 R 中，结果寄存器 R 的内容必为 0。

执行地址 1 指令，因为这时 10.00 的内容已为 1，故不管 R 内容是什么，执行它后 R 的内容必为 1。

执行地址 2 指令，这时，继电器 0.01 的内容为 0，它的非为 1。1 与 1 相与仍为 1。故执行它后，R 的内容也还为 1。

执行地址 3 指令，把结果寄存器 R 为 1 的值，赋值给输出继电器 10.00，故执行它后，输出继电器的内容必为 1。

执行地址 4 指令，用户程序结束，开始输出刷新。10.00 为 1，输出点与 COM 接通，KM 仍得电而工作。

显然，只要情况 2 继续保持，再经历一个新的周期，这个结果也将不变。

（3）情况 3。若在经历情况 1、2 后，按钮 SB1 不合上，而按钮 SB2 合上。

经输入刷新，输入继电器 0.00 置 0，输入继电器 0.01 置 1。这时：

执行地址 0 指令，把 0.00 的内容装载到结果寄存器 R 中，结果寄存器 R 的内容必为 0。

执行地址 1 指令，因为这时 R 的内容已为 1，故不管 10.00 内容是什么，执行它后 R 的内容必为 1。

执行地址 2 指令，继电器 0.01 的内容为 1，它的非为 0。1 与 0 相与为 0，故执行它后，R 的内容为 0。

执行地址 3 指令，把结果寄存器 R 为 0 的值，赋值给输出继电器 10.00，故执行它后，输出继电器的内容必为 0。

执行地址 4 指令，用户程序结束，开始输出刷新。10.00 为 0，输出点与 COM 分断，KM 失电、停止工作。

显然，只要情况 3 继续保持，再经历一个新的周期时，只是 10.00 变为 0。它改变的只是地址 1 指令的操作数。而这个操作数的改变，并不改变原来的结果。

（4）情况 4。若在经历情况 1、2、3 后，按钮 SB1 不合上，而按钮 SB2 也松开。

经输入刷新，输入继电器 0.00 置 0，输入继电器 0.01 置 0。这时：

执行地址 0 指令，把 0.00 的内容装载到结果寄存器 R 中，结果寄存器 R 的内容为 0。

执行地址 1 指令，因为这时 R 的内容已为 0，0 与 0 或，结果还是 0，故执行它后，R

的内容还为 0。

执行地址 2 指令，继电器 0.01 的内容为 0，它的非为 1。0 与 1 相与为 0，故执行它后，R 的内容也为 0。

执行地址 3 指令，把结果寄存器 R 为 0 的值，赋值给输出继电器 10.00，故执行它后，输出继电器的内容必为 0。

执行地址 4 指令，用户程序结束，开始输出刷新。10.00 为 0，输出点与 COM 分断，KM 仍失电、停止工作。

显然，只要情况 4 继续保持，再经历一个新的周期，这个结果也将不变。

不难想象，这里的情况 4，即为工作的初始状况。

总之，不停地交替运行系统程序（操作系统）及用户程序，并及时地使 I/O 状态予以物理实现，这就是 PLC 实现控制的过程。

1.2.3　可编程序控制器响应时间计算

从可编程序控制器工作过程可知，PLC 的输出对输入的响应是有滞后的。这滞后时间也称为响应时间。以扫描方式工作为例，其响应时间计算如图 1-15 所示。

从图 1-15 可知，此时间应为 t_1、t_2、t_3、t_4、t_5 与 t_6 之和。

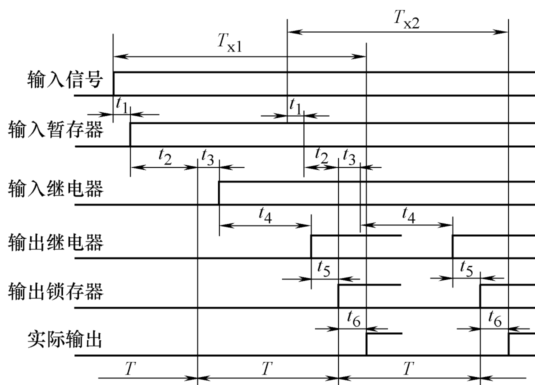


图 1-15　输出响应时间简图

这里：

t_1 为输入响应时间，从输入信号产生到输入暂存器完成存储所经历的时间。它消耗在输入电路上。可设定，默认值为 8ms；特殊的还可设定为可读取作用时间很短的信号。

t_2 为等待输入刷新时间，从输入暂存器完成存储到 PLC 开始执行输入刷新的时间。在输入暂存器完成存储时，正好是赶上输入刷新，则此时间为 0；在输入暂存器完成存储时，正好是 PLC 刚完成输入刷新，则此时间为一个扫描周期 T。

t_3 为输入刷新时间：把输入暂存器的状态读入输入继电器，即用于输入刷新的时间。

t_4 为程序执行时间：运行用户程序及公共处理时间。

t_5 为输出刷新时间：把输出继电器的状态传送给输出锁存器，即用于输出刷新的时间。

t_6 为输出响应时间：从输出锁存器状态到实际输出产生的时间。它消耗在输出电路上，取决于使用的输出电路及负载，一般为若干毫秒。

而 t_3、t_4、t_5 之和为扫描周期 T。

图中画出 T_{x1} 与 T_{x2} 两个响应时间，它们所差的是等待时间 t_2 不同。t_2 最小值为 0，最大值为 T。

所以，最长响应时间为

$$T_{x-\max} = t_1 + 2T + t_6$$

一些重要信号，如响应时间太长，则应采用中断方式处理。中断方式处理不包含 t_3、

t_4、t_5，不受扫描周期影响，而且 t_1 也可设得很短。所以可得到快速响应。

提示：在这些时间中，最长的是程序执行时间 t_4。而在 t_4 中，真正用于运行用户程序的时间约占 t_4 的 80%。

1.2.4 可编程序控制器实现控制的方式

PLC 不断地重复运行程序实现控制，称为扫描方式。此外，还有中断方式。在中断方式下，需处理的控制先请求中断，被响应后，PLC 的 CPU 停止正在运行的程序，转而去处理有关中断服务程序。待处理完中断，又返回运行原来的程序。哪个控制急需处理，哪个控制就去请求中断。哪个不需处理，将不被理睬。显然，中断方式与扫描方式是不同的。

中断方式也可称为事件触发方式。有了事件发生，即去处理有关的事件处理程序。否则，PLC 处于待机状态。

在中断方式下工作，计算机资源能得到充分利用，紧急的任务也能得到及时处理。但是，如果在一个时间内，同时有若干个中断请求，怎么办？为此，就要对中断划分等级。根据任务紧急或重要程度的不同，赋予不同的等级。显然，这就复杂了。特别是完全都用中断方式工作，就更复杂了。

较好的办法是用扫描加中断，大量控制都用扫描方式处理，个别急需的用中断处理，这样既可照顾全局，又可应急处理个别紧急或重要的事件，所以目前 PLC 用的几乎都是这种方式。而且由于 PLC 工作速度的不断提高，新推出的 PLC 已越来越增强这个中断功能。可实现的中断的事件已多到几十，以至几百种。

除了中断，还可用立即 I/O 刷新的方法加速对输入信号的响应。立即 I/O 刷新含义是：PLC 在执行程序当中，个别需要即时读入的信号，执行输入刷新指令及时读入；个别需要即时输出的信号，执行输出刷新指令及时输出，而不一定非等到 I/O 刷新时才作这种输入、输出。事实上，中断往往与刷新并用，可使中断得到更快的输出响应。

此外，可编程序控制器还可配置成多 CPU 系统。除了用于常规控制的主 CPU 外，还有辅 CPU。辅 CPU 用以处理一些特殊的（如高速计数、运动控制、过程控制、通信管理等）工作。

提示：扫描加中断、立即刷新与多 CPU 配置，加上 PLC 工作速度的提高，当今先进的 PLC，在毫秒时间内实现对外部信号的响应，检测到每秒几万、几十万赫频率的脉冲信号，已是可能了。

PLC 的实际工作过程比这里讲的还要复杂一些，分析其基本原理，也还有一些理论问题。但简单地讲，大体上就是：在空间上，由 I/O 电路进行输入输出变换、物理实现；在时间上，扫描方式运行程序，并辅以中断、立即刷新、多 CPU 处理。弄清了它，也就好理解 PLC 是怎样去实现控制的，也就好把握住 PLC 工作的要点了。

当然，由于 PLC 技术的快速发展，PLC 的工作过程与方式也会有所变化，这也是与时俱进吧！

1.3 可编程序控制器类型

关键词：I/O 点数、模拟量输入模拟量输出路数、箱体式、模块式、小型机、中型机、大型机、欧美产 PLC、日产 PLC、国产 PLC、OMRON PLC

可编程序控制器类型很多，而且越来越多，可从不同的角度对 PLC 分类。

1.3.1　按控制规模分

PLC 大致可分为微型机、小型机、中型机、大型机及超大型机。大型机、超大型机控制规模大、功能强、性能高，价格也高；而微型机、小型机控制规模小、功能也差些、性能也低些，但价格便宜。这里划分的惟一依据是控制规模。

（1）微型机控制点数仅为几点、十几点、几十点。如 OMRON 公司的 SP10、16、20 PLC，分别只能控制 10、16、20 点。OMRON 的 ZEN 机，主机有 8 点、10 点两种，还有扩展，最多可扩到 34 点。这个机型，有的还内嵌有简易的编程工具，便于编程。由于它的价格低廉、使用方便、工作可靠、体积很小……而且它的可输出电流比其它 PLC 都大，有的可达 8A，因而可以成为继电器控制的替代品，也因此有的则称其为可编程继电器（PLR）。

（2）小型机控制点可达 100 多点，甚至更多。如和利时公司的 G3 机，控制点数可达 256 点。再如 OMRON 公司称之为紧凑型 PLC，型号有 CPM1A（最大 I/O 点数为 100 点）、CPM2A（最大 I/O 点数为 120 点）、CPM2AE（最大 I/O 点数为 120 点）、CPM2C（最大 I/O 点数为 362 点）、CPM2AH（最大 I/O 点数为 120 点）、CPM2AH-S（最大 I/O 点数为 356 点）等。

此外，还有 CP1H（最大 I/O 点数为 320 点）、CP1L（最大 I/O 点数为 160 点）及 CJ1M（最大 I/O 点数为 640 点）机。其性能、功能都比紧凑型机又有了提升。

（3）中型机控制点数可达近 500 点，甚至千点。型号有 C200Hα（最大 I/O 点数为 1184 点）、CJ1/CJ1H（最大 I/O 点数为 2560 点）。

（4）大型机控制点数一般在 1000 点以上。控制点数可达 1024 点（数字量）、256 点（模拟量）。如 OMRON 公司早期的 C1000H、CV1000 机，当地配置可达 1024 点。C2000H（可热备）、CV2000 当地配置可达 2048 点。加上远程配置这个点数还可增加一倍。还有 CV、CVM1、CVM1D（可热备 ）也是大型机。而目前主推的是 CS1（最大 I/O 点数为 5120 点）、CS1D（可热备、最大 I/O 点数为 5120 点）机。

（5）超大型机控制点数可达万点，甚至几万点、十几万点、几十万点。有的 PLC 可在主机架上可安装多个 CPU 单元，以及它是按全新的控制方式工作，其控制点数已不受什么限制。

应该讲，以上这种划分是不严格的，只是大致的，所介绍的情况也是会有变化的。划分的目的只是帮助读者建立控制规模的概念，以便于日后进行系统配置及使用。一般讲，根据实际的 I/O 点数，凡落在上述不同范围内，选用相应的机型，性能价格比要高；相反，肯定要差些。

自然，也有特殊情况。如控制点数不是非常多，不是非用大型机不可。但因大型机的特殊控制单元多，可进行冗余配置，因而采用了大型机。

1.3.2　按结构特点分

可分为一体化、模块式及内插板式。此外，瑞士思博（Saia-burgess）公司生产的，从小型、中型的 PCD1、PCD2、PCD3 到大型的 PCD4、PCD6 系列产品，结构很特别，算是另类。

1. 一体化 PLC

也有称之为箱体式的 PLC。它把电源、CPU、内存、I/O 系统都集成在一个小箱体内。一个主机箱体就是一台完整的 PLC，就可用以实现控制。微型、小型机多为箱体式。主箱体也称为 CPU 模块。

为了系统配置方便，有的主箱体还可增加内插选件，如通信接口选件、内存选件、模拟量输入输出选件等，为要增强主箱体功能的用户提供了方便。

此外，还有扩展箱体（模块）。扩展箱体外观与主箱体类似，但一般只有 I/O 系统及电源（有的没有，其电源可由主箱体提供），有的另有其它功能。如果主箱体控制点数不符合需要或功能不足，可再接若干个扩展箱体。

2. 模块式 PLC

它由具有不同功能的模块组成。其主要模块有 CPU 模块、输入模块、输出模块、电源模块、通信模块和机架等。超大、大、中型机都是模块式的。

从结构上讲，由模块组合成系统有 3 种方法：

（1）无底板，靠模块间接口直接相连，然后再固定到相应导轨上。

（2）有底板，所有模块都固定在底板的安装槽上，比较牢固。

（3）用机架代替底板，所有模块都固定在机架上的相应格子中。这种结构比底板式的复杂，但更牢靠。

3. 内插板式 PLC

为了适应机电一体化的要求，有的 PLC 制成内插板式的，可嵌入到有关装置中。它有输入点、输出点，还有通信口、扩展口及编程器口。普通 PLC 有的功能，它也都有。但它只是一个控制板，可很方便地嵌入到有关装置中。

1.3.3 按生产厂商分

（1）欧美产 PLC 主要有：德国的西门子、美国的 AB（Allen-Bradley，现已被美国的 Rockwell 自动化公司收购）、美国 GE（与日本 FANUC 合资的 GE FANUC）、法国的施耐德（原美国 MODICON 公司已被其收购）及瑞士 ABB 等公司的 PLC 产品。

（2）日产 PLC 主要有：日本的欧姆龙（OMRON）、三菱、松下、东芝、富士、光洋、日立等公司的 PLC 产品。

欧美日产品都经过多年、多代的发展。品种齐全、性能高、质量稳定，但价格很高，特别是欧美产品价格更高。

（3）韩国产 PLC 主要有 LG。中国台湾产 PLC 主要有：台达（也称中达）、永宏等公司的 PLC 产品。它们多是近 10 来年才发展起来的。品种不太全、性能不是太高、质量也有待考验，但价格较便宜。

（4）国产 PLC 主要有和利时、特维森等公司的 PLC 产品。国产 PLC 的开发起步较晚，但采用了正确的迎头赶上的战略。如和利时的 G3 小型机，就是利用高起点的软、硬件平台自主开发的，它的功能较强，性能也较高，再进一步提高后，做到“后来居上”则是完全可能的。此外，国产 PLC 还有明显的价格、服务及可定制上的优势，因而在我国的 PLC 市场中，国产 PLC 的份额也必将逐步增加。

1.3.4　按其它特点分

（1）按电源分有：直流 24V 的、交流 220/110V 的。交流也还有宽幅（有的称为任意的，80～240V 均可）与非宽幅（220、110V 可选）。

（2）按内存分有：RAM 加电池的、ROM 的、闪存的。有另加内存卡的，也有不加的。

（3）按功能分有：普通顺序控制功能的、可编程过程控制器（Programmable Process Controllor，PPC）功能的；可编程运动控制器（Programmable Motion Controllor，PMC）功能的；以及可编程自动控制器（Programmable Automation Controllor，PAC）功能的。CJ1 系列内置回路控制板的 CPU 单元解决 CJ1 的回路控制问题，回路板有各种控制方式。

（4）按使用环境分有：普通使用环境的和环境扩展型的。

（5）按可靠性分有：普通的、冗余的及安全型 PLC（Guard PLC）。

（6）按是否为标准产品分有：标准产品和可定制产品。所谓可定制产品是指用户可根据特殊需要单独定制，这样更能达到使用的要求。一般来讲，国外的 PLC 是没有定制产品的，但和利时等国产 PLC 多有此类产品，这也是国产 PLC 的一个大优势。

随着 PLC 新产品的不断出现，还可对 PLC 作不同的分类。但有了以上分类的分析，读者也许会对 PLC 有一个大致的而又比较全面的了解。

1.4　可编程序控制器组成

关键词：箱体、模块、CPU 单元、内存单元、I/O 单元、特殊 I/O 单元、通信模块、编程器

（1）从结构看。PLC 是由一些箱体或模块组成的。而这些箱体或模块则由不同功能的单元构成。

如箱体式 PLC，其主箱体（也有称为 CPU 箱体），则由 CPU 单元、内存单元、外设接口（有的还有通信接口）、I/O 单元、电源、箱体间接口及其它附件等构成；其扩展箱体则由 I/O 单元、箱体间接口、电源及其它附件等构成。

再如模块式 PLC，其 CPU 模块，则由 CPU 单元、内存单元、接口单元，有时还有 I/O 单元及接线器等构成。其输入模块，则是由多个输入电路、接线器及相应接口构成。

（2）从功能看。是由 CPU 单元、内存单元、I/O 单元（有的称为信号单元）、特殊单元（有的称为功能单元）、外设与通信接口及电源等组成。图 1-16 所示为功能组成原理图。有

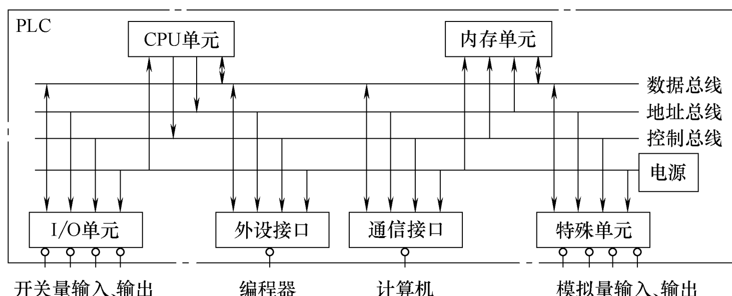


图 1-16　PLC 组成原理图

时，也可把上述单元改称为模块。

从图 1-16 可知，这些单元工作全由电源供电。它们之间全是通过总线进行联系。用地址总线建立地址联系，再用数据总线传送数据。而这两个总线则都受 CPU 管理的控制总线控制。

如 PLC 进行 I/O 刷新，则 CPU 控制器管理的控制总线，使 I/O 单元与内存的 I/O 区建立地址联系，并在所联系的地址间有步骤地进行数据交换。

再如执行用户程序，则 CPU 控制器管理的控制总线，使 CPU 的指令寄存器与内存的用户程序存储区建立地址联系，然后把将要执行的指令传送给这个寄存器。进而，经指令译码、执行，再把结果送到与内存相应的内部器件区中。

此外，PLC 还提供外设、通信接口，可用以连接 PLC 编程器、计算机、其它 PLC 或智能装置。

1.4.1 CPU 单元

CPU 主要含有运算器、控制器、寄存器。不管什么样的 CPU，其实质都是同步时序电路，是分周期同步工作的。为了支持它的工作，当然要向其提供振荡信号；同时，还要向其提供工作电源。

CPU 的控制器控制 CPU 工作及控制总线，如指令的读取、解释及执行就是靠它控制的，但它的工作节奏由振荡信号控制。

CPU 的运算器用以进行数字或逻辑运算，在控制器指挥下工作。

CPU 的寄存器存储参与运算的数据，并存储运算的中间结果。它也是在控制器指挥下工作。

CPU 虽划分有以上几个部分，但在物理上却是集成在一个芯片上的。按原来意义的晶体管计，一个 CPU 芯片集成了几万、几十万，甚至几百万个晶体管。有的芯片还集成了 ROM、串行口、并行口等。

PLC 的 CPU 芯片实际就是微处理器（Micro Process Unit），或干脆就是单片机或嵌入式计算机。只是它为专用于 PLC，并多为厂家自行开发。这样既可以达到最佳的性能匹配，又可以实现商业保密。

也有的 PLC 用的芯片就是通用的单片机，只是内部装有自编程的监控程序，并靠这个监控程序，使其实现 PLC 的功能。

由于这些电路是高度集成的，对 CPU 内部的详细分析已无必要。特别对使用者而言，完全可把它看成为“黑箱”，只要弄清它的功能与性能，能正确地使用也就够了。

1.4.2 内存单元

PLC 的内存每个存储单元的长度也是固定的。西门子 PLC 以字节计。OMRON、三菱以字（或步）计。其种类是较多的，具体有：

（1）依用途分，PLC 内存有：

用于存储系统程序（操作系统）的内存；

用于存储工作数据的内存；

用于存储系统设定的内存；

用于存储用户程序的内存；

用于种种备份的内存。

尽管这么划分，但除了备份外，其地址是统一的。而对于用户，只有系统工作单元（包括 PLC 的内部器件、系统设定区及用户程序区）。

（2）依介质分，PLC 的内存有：

只读存储器（ROM）或电可编程只读存储器（EPROM）用于存储系统程序（操作系统）。对于早期的 PLC，用户程序定型后，也可存于这种类型的存储器中。这种内存不掉程序，免维护。

电可擦可编程只读存储器（EEPROM）在程序写入后，掉电也不丢失程序，并且可多次改写程序，少的几百次，多的几乎不受限制。近来这种内存用得很多，且多用于存储用户程序。

闪烁（Flash）式内存，可多次改写程序，掉电后也不丢失程序。CPM1A 机用的即为这种内存，其容量可达 2k 字。

随机存取存储器（RAM）的访问速度快，可任意改写，但要有电池支持，多用于存储数据及用户程序。

（3）依内存的分布分，有：

主内存：是内存的主体，而且也是 CPU 直接访问的对象，早期的 PLC 多数只有这种内存。

辅助内存：只用以存储用户程序或其它数据的备份，一般为内存卡，也是闪烁存储器。它多是选件，也可不用。在主辅内存间，可通过相应操作或通过用户程序交换数据。

此外，一些功能很强的模块往往自身也带内存，只是对这些内存，用户可不必去处理它。

1.4.3　I/O 单元

I/O 单元是集成了 I/O 电路的单元，类型、规格很多。图 1-17 所示为一个输入单元的电路原理图。从图 1-17 可知，当输入点 IN 00（IN01 等未画出）ON（如本例加上 24V 电压）时会在该输入回路上产生接近 8mA 的电流，使输入发光二极管亮，经光的耦合，使与内部电路对应的输入寄存器置 1。反之，若 IN 00　OFF（去除所加电压），则置 0。从图还可知，输入用的电源正负极可任意连接。

图 1-18 所示为一个输入单元的接线参考图。该输入单元为 16 个输入点，并只有两个公用回路（COM 端）。

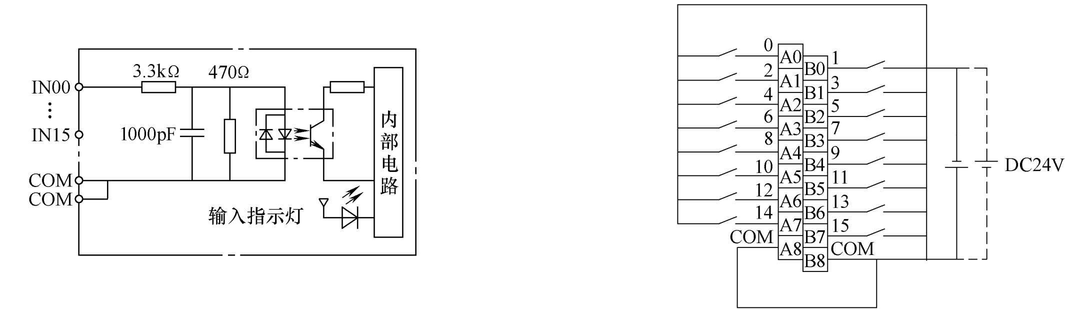


图 1-17　输入电路原理图　　图 1-18　输入单元接线参考图

图 1-19 所示为输出单元的电气原理电路。图 a 为继电器输出，从图可知，其继电器线圈与内部电路（输出锁存器等）相接，而继电器触点则直接用于接用户电路，这里的继电器还起到 PLC 与外部电路隔离的作用。图 b 为晶闸管输出。图 c、d 为晶体管输出。它们的内部电路与输出，在电路上是隔离的，但通过光的耦合建立联系。

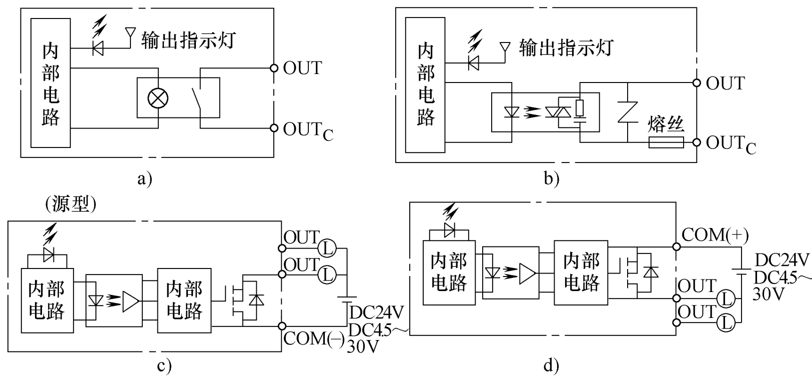


图 1-19 输出电路原理图

输出电路可通过的电流受继电器触点容量的限制，一般为 1～2A，并依负载而定。感性负载时，可通过的电流要小些。

半导体输出有源型与漏型两种，以便驱动相应直流负载。它的响应速度快，但可通过的电流小。

晶闸管可通过的电流大，只能用于交流负载。其响应速度要比继电器快，但当从 ON 到 OFF 时，比晶体管慢。

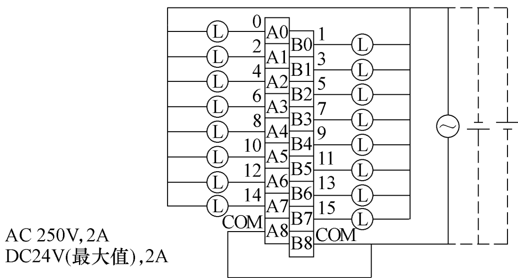


图 1-20 输出单元接线参考图

图 1-20 所示为一个继电器输出单元的接线参考图。该图的 L 为负载，COM 为公用端。因为 PLC 只提供触点，故外电路用直流或交流均可。输出模块的点数也有多种规格，如 12 点、16 点、32 点等。

1.4.4 其它单元

（1）特殊 I/O 单元。主要有用以处理模拟量输入的模块、用以处理模拟量输出的模块、用以处理温度输入的模块、用以控制温度的温度控制模块、用以处理脉冲量输入高速计数模块、用以处理脉冲量输出的位置控制模块、用以既可处理脉冲量输出又可处理模拟量输出的运动控制模块等，而且随着 PLC 技术的发展，这些模块的功能在不断增强，类型在不断增加，性能在不断提高。

（2）CPU 总线模块。主要是用以组网的网络模块，具体与组建的网络有关。常用的有以太网模块、Control Net 模块、Device Net 模块、Profibus 模块等。此外有一些功能很强的特殊 I/O 模块，OMRON 称之为 CPU 总线单元。

在 PLC 的 CPU 单元上，一般都配置有 USB 口或外设口，以及 RS232C、RS442A 或

RS485 接口或选件。此外，有的 PLC 还可以另外配置标准串口模块。其功能是与编程器或计算机通信，或与其它 PLC 及智能设备通信。

（3）电源模块。PLC 电源有交流与直流两种。直流多为 24V，多用于移动环境。交流有宽幅的 110V、220V 均可。同时，还多提供有 24V 的直流电源，以提供给开关量输入电路使用。而非宽幅的电源，使用时一定要小心，切不可把应接 110V 的接到 220V 的电源上，也不可把接直流电源的接到交流电源上。由于这样的错接，而出现烧电源的事例在实践中是不少见的。

此外，还有 I/O 控制与接口模块、连接电缆，对有底板或有机架的 PLC，还要有底板或机架等。

1.4.5　外部设备

主要用于进行 PLC 编程，数据的读入、显示、存储及打印。具体有四大类：

（1）编程设备。简单的为简易编程器，多只接受助记符编程，个别的也可用图形编程（如日本东芝公司的 EX 型 PLC）。复杂一点的有图形编程器，可用梯形图语言编程。

（2）监控设备。小的有数据监视器，可监视数据；大的还可能有图形监视器，可通过画面监视数据，除了不能改变 PLC 的用户程序，编程器能做的它都能做。再就是通用的人机界面，可用图形或数字显示或输入与 PLC 交换的数据，实施对 PLC 的监控。性能好的 PLC，这种外部设备已越来越丰富。OMRON 公司生产人机界面，与自身 PLC 连接，还具有简易编程器的功能。

（3）存储设备。用于永久性地存储用户数据，有存卡器、磁带机、软盘或 ROM 写入器，以及相应的接口部件。各种 PLC 大体都有这方面的配套设施。

（4）输入输出设备。如条码读入器、小型打印机等。

从某种意义上说，最一般的 PLC 外部设备可以说是个人计算机，PLC 配有通信口，计算机安装上相应的软件，两者链接后，即可用计算机实现 PLC 程序的编制、调试、存储，也可实现 PLC 数据的显示、存储及打印。

外部设备已发展成为 PLC 系统的不可分割的一个部分。但它不是 PLC 工作所必备的。

提示：这里介绍的只是 PLC 的一般组成。 PLC 的具体组成取决两个因素：一是 PLC 的品牌、机型，不同的品牌、机型 PLC 组成的“可能”是不同的；二是用户的配置，即用户在这个“可能”中可做不同的选择。

1.5　可编程序控制器特点

关键词：信息、功能丰富、使用方便、工作可靠、经济合算

PLC 的输入与输出在物理上是彼此隔开的，其间的联系主要通过程序实现。其工作基础是信息流，而不是物流或能量流。

信息不同于物质与能量，便于处理、便于传递、便于存储，还可共享、重用。正是信息的这些特点，决定了 PLC 具有功能丰富、使用方便、工作可靠及经济合算四大特点。

1.5.1 功能丰富

系统功能与系统组成紧密相关。PLC 丰富的功能与 PLC 拥有快速工作的 CPU、各种各样的 I/O 接口及网络接口、大容量的内存、可靠与成熟的操作系统相关。正是 PLC 具有类似计算机那样的组成，所以其功能是非常丰富的，具体功能有：

逻辑处理功能；

数据运算功能；

准确定时（高档 PLC 可进行毫秒级计时）功能；

高速计数（高档 PLC 计数频率可高达几百 kHz）功能；

中断处理（可实现种种内外中断、高档 PLC 可实现毫秒级响应）功能；

程序与数据存储功能；

联网通信功能；

自检测、自诊断功能；

等等。

因此，PLC 既能用于实时控制，又能用于数据运算；既能处理现场信息，又能实现远程监控。凡普通小型计算机以及其联网所能做到的，它也都可以做到。

像 PLC 这样，集成这样丰富的功能于一身，是别的电控制器所没有的，更是传统的继电控制电路所无法比拟的。

丰富的功能为 PLC 的广泛应用提供了可能；同时也为工业系统的自动化、远程化、信息化及智能化创造了条件。

1.5.2 使用方便

用 PLC 实现对系统的控制是非常方便的。这是因为：

首先，PLC 控制逻辑的建立是程序，用程序代替硬件接线。编写程序比接线、更改程序比更改接线，当然要方便得多！

其次，PLC 的硬件是高度集成化的，已集成为种种小型化的箱体或模块，而且这些箱体或模块是配套的，已实现了系列化与规格化。种种控制系统所需的箱体或模块，PLC 厂商多有现货供应，市场上可方便购得，所以硬件系统配置与建造也非常方便。

正因如此，用 PLC 才有这个“可”字。对软件讲，它的程序可编，也有办法编。对硬件讲，它的配置可变，而且也易于变。

具体地讲，PLC 有 5 个方面的方便：

（1）配置方便。可按控制系统的需要确定要使用哪个商家的，哪种类型的 PLC，以及用什么箱体或模块，要多少箱体或模块。确定后，到市场上订购即可。

（2）安装方便。PLC 硬件安装简单，组装容易。外部接线有接线器，接线简单，而且一次接好后，更换模块时，把接线器安装到新模块上即可，可不必再接线。内部什么线都不要接，只要作些必要的 DIP 开关设定或软件设定就可工作。

（3）编程方便。PLC 编程可以用编程器，也可以用计算机。使用的编程语言可以是梯形图语言，也可以是助记符。有的还可用高级语言，如 BASIC 语言、C 语言，而且厂商提供有编程软件，为 PLC 编程提供了方便。

在调试程序方面，有的厂商的 PLC 还有计算机仿真软件，可在 PLC 没到货时就可进行程序调试。

PLC 的程序便于存储、移植及再使用。某定型产品用的 PLC 的程序完善之后，凡这种产品都可使用。这比起继电控制电路设备台台都要接线、调试，要省事及简单得多。

由于以上的方便，用 PLC 作控制系统集成，其开发过程比过去用继电器要快得多，小系统所用的时间几天就够。大系统要多用一些时间，但与 PLC 有关的也不多，主要用于其它相关的配置上。

（4）维修方便。这是因为：

1）PLC 工作可靠，故障不多，减轻了维修的工作量。

2）维修方便。PLC 都设有很多故障提示信号，而且 PLC 可作故障记录，所以出了故障，很易查找与诊断，同时排故也很简单，可按箱体或模块排故，而箱体或模块的备件市场上可以买到，更换就可以。至于软件，调试好后是不会出故障的，至多也只是依据使用经验使之完善就是了。

（5）改用方便。PLC 用于某设备，若这个设备不再使用了，其所用的 PLC 还可给别的设备使用，只要重新接线及改编程序就可办到。如果原设备与新设备差别较大，它的一些箱体或模块也还可重用。

1.5.3　工作可靠

用 PLC 实现对系统的控制是非常可靠的。在硬件与软件两个方面它都有很多有效的措施：

1. 在硬件方面

（1）对输入信号多作了滤波。而且，输入输出电路与内部 CPU 是电隔离的。其信息靠光耦器件或电磁器件传递。同时，CPU 板还有抗电磁干扰的屏蔽措施，可确保 PLC 程序的运行不受外界的电磁干扰。

有很多这样的实例，原来用计算机的采集卡采集数据，因干扰大而无法正常工作，而改换用 PLC 后，顿即可正常工作。

（2）PLC 使用的元器件多为无触点的，而且为高度集成的，数量并不太多，也为其可靠工作提供了物质基础，而且所用的元器件都经严格监测、老化与筛选，质量是有可靠保证的。其输出用的继电器虽为有触点的，但它的触点是在密封的真空条件下，故其寿命也可达几十万次。

（3）在机械结构设计与制造工艺上，可确保 PLC 耐振动、耐冲击；使用环境温度可高达 50℃以上，有的 PLC 可高达 100℃；有的在低温，低到零下 40 ~ 50℃，还可正常工作。

（4）有的 PLC 的模块可热备，一个模块工作，另一个模块也运转，但不参与控制，仅作备份。一旦工作模块出现故障，热备的可自动接替其工作。还有更进一步冗余的，采用三取一的设计，那就更加可靠了。

2. 在软件方面

（1）PLC 的工作方式一般为扫描加中断，这样既可保证它能有序地工作，避免继电控制系统常出现的“冒险竞争”，其控制结果总是确定的；而且又能应急处理急于处理的控制。

（2）为监控 PLC 运行程序是否正常，PLC 都有“看门狗”（Watching dog）监控程序。运行用户程序开始时，先清“看门狗”定时器，并开始计时。当用户程序一个循环运行完了，则查看定时器的计时值，若超时（一般不超过 100ms），则报警；若不超时，则重复起始的过程，PLC 正常工作。显然，有了这个“看门狗”监控，可保证用户程序的正常运行。

（3）PLC 还有很多防止及检测故障的指令，可通过编制相应的用户程序，对 PLC 的工作状况，以及 PLC 所控制的系统进行监控，以确保其可靠工作。

（4）PLC 每次上电后，还都要运行自检程序及系统初始化。如出现故障也有相应的出错提示。

正是 PLC 在软、硬件诸方面有相应措施，才保证 PLC 具有可靠工作的特点。它的平均无故障时间可达几万小时以上；出了故障平均修复时间也很短，几小时甚至几分钟即可修复。

在国外，曾有人做过为什么要使用 PLC 的问卷调查。在回答中，多数用户把 PLC 工作可靠作为选用它的主要原因。

在国内，多年使用经验也说明，PLC 工作是非常可靠的。正使用的 PLC 往往不是由于用坏而被淘汰，而往往是由于 PLC 技术发展太快，显得落后而被淘汰。

1.5.4　经济合算

经济是基础，效益是生命。高新技术的生命力就在于，它能给社会带来巨大的社会效益与经济效益。说科技是第一生产力，根本原因也就在于它能给社会带来效益。PLC 的产生、发展以至于今天的红火和未来的辉煌也在于此。

经验证明，尽管使用 PLC 首次投资要大些，但从全面及长远看，使用 PLC 还是合算的。这是因为：使用 PLC 的投资虽大，但它的体积小，所占空间小，辅助设施的投入少；系统集成方便，建造周期短；使用时省电，运行费用少；工作可靠，停工损失少；维修简单，维修费用少；还可再次使用以及能带来附加价值等等，从中可得更大的回报，所以在多数情况下，它的效益是可观的。使用 PLC 与使用传统的继电控制装置相比，其效益可按下式计算：

$$V = C - T$$

式中　V——使用 PLC 可能带来的效益；

C——使用 PLC 所能带来种种产出，而 C 可由下式计算：

$$C = C_1 + C_2 + C_3 + C_4 + C_5 + C_6$$

其中　C_1——使用 PLC 后控制系统安装费的节省（包括材料费、占地费及工程周期与工时的节省）；

C_2——使用 PLC 后节省的维修费（包括备件费及维修工时费的节省）；

C_3——使用 PLC 后节省的运行费；

C_4——使用 PLC 后减少的费用系统停工损失（由停工工时的减少与单位时间系统所能创造的效益的乘积确定）；

C_5——使用 PLC 后所能带来的附加值；

C_6——PLC 到寿命后可回收的残值。

T——使用 PLC 所需增加的投入，这里的 T 由两项组成，即

$$T = T_1 + T_2$$

其中　T_1——使用 PLC 时一次性多投入的资金（注意这里的多字，它突出了相对数）；

T_2——多投入资金的占用费，T_2 可由下式计算：

$$T_2 = T_1 \cdot K \cdot n$$

其中　T_1——含义同上；

K——占用资金的年利率；

n——PLC 使用寿命，以年为单位。

显然，经计算或估算后，若 V 为正值，则使用 PLC 是合算的。

V 为正值的关键在于 C_4、C_5。特别是随着控制系统的复杂程度的增加，使用 PLC 不仅出于方便与可靠的需要，而且带来的经济效益也是明显的。

图 1-21 为 T 与 C 随系统复杂程度（以控制点数计）变化的情况。由图可知，在相当多的控制点数之后，由于 C 增加要比 T 增加快得多，所以 V 将出现正值。

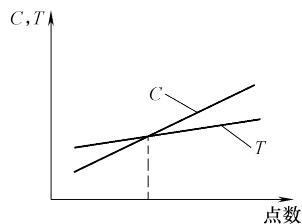


图 1-21　T、C 随点数变化示意

使用 PLC 与使用其它高新控制系统在经济效益上也具有优势。有人做过分析，一个相当规模控制系统，同样用 GE_ FANAC 公司的产品，一个为 PLC，另一个为 DCS（集散控制系统），前者可比后者节省 1/3 的投入。

总之，PLC 具有功能丰富、使用方便、工作可靠及经济合算的特点，使得它既非常有用，又非常好用、耐用、省用，有无限的生命力和广泛的应用前景。短短 30 多年，它从诞生、发展、成熟到不断完善，已成为工业自动化的支柱，并发展成为强大科技产业。可以这么说，在当代，一个工业控制系统，或较先进的工业产品，其控制装置若不使用 PLC，那是不可想象的。

1.6　可编程序控制器使用

关键词：系统配置、基本配置、扩展配置、特殊配置、冗余配置、联网配置、附加配置、PLC 编程、分配 I/O

要使用好 PLC，首先要做的工作有两个：

一是进行系统配置；

二是编制用户程序。

这涉及 PLC 使用的硬件与软件两大方面。

1.6.1　系统配置

PLC 系统配置也就是 PLC 硬件系统构成。如果不考虑传感器、执行机构，就是若干模块、箱体及有关附件，箱体、模块又各有其型号与规格。根据工艺要求，选哪些，各选多少，怎么组合成系统，就是这里要讲的系统配置。

系统配置是问题的综合，与问题的分析不同，同样题目，答案可能很多，所以系统配置的重要工作就是要从多个答案中优选一个方案。

系统配置要遵守一些原则，要用科学的方法，按步骤进行。

1. 系统配置原则

（1）完整性原则：PLC 加被控制对象可看成是一个 PLC 控制系统。PLC 又是这系统的一个子系统。按系统论的观点看，凡系统，要使其发挥应有作用，系统必须完整，缺东少西，系统将无法工作。所以在对 PLC 进行系统配置时，首先要考虑系统是否完整，不能“丢项”，该配的都得配齐；如果丢项，在安装或使用时再追加，将带来诸多麻烦，同时还会延误工期。

完整性要突出在全面考虑上，既要考虑 PLC 系统，又要考虑 PLC 所控制的系统。依照经验至少要弄清以下几个问题：

1）CPU 模块配置得如何？其性能合乎要求否？特别是工作速度及控制规模要合乎要求。

2）电源模块如何？电源有两个概念：一是 PLC 自身所用的电源，是用交流？还是直流？什么规格？在 PLC 使用的环境中有无此电源？要是没有，怎么办？……另一是 PLC 为输入电路提供的电源，如直流 24V 电源，有没有？多大容量？够不够用？也要测算或估算，不够又怎么办，也要作安排。有时，对一些特殊模块还要求有特殊的电源。

3）内存怎么样？新型的 PLC 一般都装有 RAM，并有电池支持。所以，从理论上讲，用户可不再配置。但为了程序安全，一般还可配置 EEPROM 或 ROM，甚至于内存卡。OMRON 公司新出品的 CPM1A 机可存 2K 字的用户程序，用的是新型存储器，既不丢程序，还不用电池。另外，有的 PLC 使用时还要具备系统时钟。这时，使用的内存有的还有特殊要求（CPU 也有相应要求），这也是要考虑的。选内存时，重要指标是容量，选用时一定要满足要求。

4）I/O 模块、特殊模块，也要按系统规模配置，只是要注意有的模块有特殊要求。如有的要另加电源，如 OMRON 公司 CQM1 机的模拟量单元，就要另配模拟量单元电源。有的要多占槽位的，如 OMRON 公司的位控单元就要占两个槽位。有的还只占半个槽位，如西门子公司有的 I/O 模块，等等。这些也都要考虑到。

5）机架、连接单元和连接电缆以及必要的附件。不同商家、不同型号的 PLC 情况不一，虽不那么重要，但有时缺了也很耽误事。这些选项比较零散，所以在配置时要格外小心。弄得不好，即使少一根电缆，也将损害 PLC 系统的完整性，从而影响 PLC 系统工作。

6）外设的配置也不可忽视。如果计划用计算机编程，那也要配置编程软件。

（2）可靠性原则：不少厂商选用 PLC，其出发点首先是因为 PLC 工作可靠。为此，PLC 系统配置当然要注意这个问题。

可靠性可从四个方面考虑：一是 PLC 自身产品质量；二是供货方的技术服务；三是重要场合下的可靠性要求；四是冗余配置。

1）PLC 自身质量。生产 PLC 的厂商很多，质量、性能及价格各异。但一般来讲，性能好、质量高的，价格总是高点，因此作系统配置时要作必要的权衡。

性能自然要尽可能好，但以满足要求或略高为度。而质量则无论如何要有保证，要用正品，要用经质量论证确认为合格的产品。为此，大公司的产品，或有较长生产经验的厂商的产品是值得推荐的。

2）PLC 厂商的技术服务情况：PLC 产品质量再高总还会有故障的，只是发生故障的平均时间间隔长些。而出了故障怎样减少排除故障时间，也是可靠性要考虑的另一个问题。

一般来讲，PLC 厂商及其在当地的代理商技术服务情况与减少故障排除时间是有很大关系的。技术服务好、配件多、供货及时，则便于用户查找故障，购置配件，当然可以减少排除故障时间。除了考虑 PLC 厂商，也要考虑厂商生产的型号。所选的产品型号也要考虑到尽可能选用有完整的技术资料、在当地配件较多的型号。

3）选用有特殊可靠措施的 PLC：为了系统工作可靠，不少厂商推出了有特殊可靠措施的 PLC。如果可靠性要求较高的系统，则可选用这种 PLC。

各个厂商有各自针对不同方面的特殊措施，并多已成为自己的技术专利。如西门子公司在接地防干扰方面有不少特点，一些 I/O 模块可接地，有的可悬浮，可满足不同的要求。GE—FANAC 公司生产的 90-70 系列机，有的 I/O 模块可监测到用户的输入信号线断线或短路的故障。若配置系统时选用它，再在编程时，对监测到的这种故障进行报警或作其它处置，则可更加有力地提高系统的可靠性。

正是因为特殊可靠措施带有专利性质，为了取得技术优势，不少厂商已开始注意了这个问题。相信将有越来越多的措施出台。真正受益当然是 PLC 的用户了。所以，如果对系统有特殊的可靠性要求，则在配置时就要考虑选用有这个相应措施的 PLC。

4）特殊可靠要求的配置：有一些重要的行业，要求配置的系统要特别可靠，不许出故障，或出了故障要在尽可能短的时间内排除故障，甚至出了故障仍可工作。这时可用 3 种方法解决：

① 冷备份。除了必需的模块都配齐，并上线运行之外，主要的、易损坏的模块要预先购置一些，作为备份。一些上线的模块出现故障，很快地可把备份换上，以减少停产损失。不少化工行业所配置的系统多为这种配置。

② 热备份。如日本 OMRON 公司就有这样的 PLC，一套 PLC 配上两个 CPU，一个工作，另一个热备份。热备份 CPU 虽也运行程序，但不实现对外控制，一旦工作的 CPU 出现故障，热备份的自动接替工作，而出故障的可脱机修理，可做到系统不停机，即可排除故障。

③ 冗余配置。热备份实质也是冗余配置，因为热备份的模块，实质是多余的。但这里讲的冗余配置比热备份更进一步。

有三冗余配置，三套模块同时工作，其结果根据三者的多数决定。这样的系统故障率比无冗余的要低得多。

还有四冗余配置，如 GE—FANAC 公司生产的一种输出模块，四套同用，其中有一套，甚至于两套出了故障，系统仍可输出。

冗余配置多用于非常重要的场合。除冷备份，其它的不是所有厂商或所有型号的 PLC 都有这种功能，所以要求作这样配置时，只好选用有相应功能的 PLC 了。

（3）发展性原则：PLC 是当今众多高科技集成的结果。高科技发展迅猛，所以 PLC 总是日新月异，年年都有新模块出台，隔年总有新品种问世。PLC 的工作可靠，自然寿命都较长，但它的技术发展很快，技术寿命并不长。PLC 往往不是使用坏了，而是还未用坏，由于技术落后了，又不得不淘汰。高科技总是越新越好嘛！

另外，当今科技的发展，各行各业的技术进步都较迅速，即使传统产业也不例外。

为此，PLC 系统配置应坚持发展性原则。发展性原则有两个重要内涵：

1）作系统配置时要留有发展的余地。这个余地有如下几个方面：

（a）系统控制规模增大了怎么办？为此，I/O 点最好略有富裕，不宜满打满算。

（b）系统控制档次提高了怎么办？如手动控制升为半自动控制，或半自动控制升为全自动控制，或自动控制范围进一步扩大，等等，如有必要，在作系统配置时应适当考虑。

（c）系统控制地域扩大了怎么办？如原为当地的系统，也可能发展成另一主站的子系统，或可能外挂自身的子站，或与上位计算机联机以实现智能化，等等。如有必要，在作系统配置时也要考虑。

总之，要考虑到发展，要使所配置的系统具有弹性，留有发展的余地。

2）对 PLC 选型时应尽可能用新型号。这个道理是明显的。选新型的 PLC，其技术寿命长，近期不易淘汰，对系统的维修、补充备件是大有好处的。不然，选用即将淘汰的、技术寿命不太长的 PLC，过几年这种 PLC 一旦被淘汰，连备件都买不到，对系统的维修是很不利的。

（4）继承性原则：俗话说，做事情要瞻前顾后。系统配置坚持发展性原则可以说是瞻前，而坚持继承性原则是顾后。

顾后就是要考虑到曾经使用过 PLC 的历史及情况。在作新的系统配置时，尽可能选用已用过厂商的产品，甚至于选用已用过的机型。

这样，PLC 的外设可以共用，如编程器就可多次使用。如果新配置的系统的机型与原有的有继承关系，新系统就可不配编程器，不就节省了吗？

再就是一些编程工具软件，若按继承性原则配置，新系统也还可再用。

还有使用经验，若按继承性原则配置，所积累的宝贵经验也可派上用场，甚至有的程序还可移植。这对缩短编程时间，以至于正确地使用 PLC 都大有好处。

应该讲，当今我国实行开放政策，一些企业引进不少 PLC，但由于忽视继承性，致使有的厂商拥有 PLC 生产厂商及型号很杂，给使用、维护都带来不便，而且在经济上也不上算，是值得汲取的教训。

（5）经济性原则：经济是一切活动的基础。进行系统配置当然要考虑经济性。经济上是否合算，应作为是否采用 PLC、要用什么样 PLC 的重要评价标准。

经济性应从两个方面考虑：

1）用 PLC 是否合算。这个合算当然要全面分析，既要考虑采用 PLC 一次性投入要小些，但用了 PLC 之后，会带来一系列效益，如可节省使用费，节省安装空间，减少系统故障停工损失，带来附加值，等等。

只要多得到的大于多投入的，就是合算，就可采用。

2）用什么样的配置更合算。这就要求系统配置时要精确掌握各个厂商的报价，在进行系统配置后要算算账，然后取其最优者。

2. 系统配置方法

系统配置可以用类比法、估算法、计算法及测试法。

（1）类比法。类比法也就是经验法。用自己的或别人的经验与所要配置的系统作类比。从类比中初步确定要选用什么厂商的 PLC，用什么型号，用哪些模块。

类比法虽不很准确，但由于有经验借鉴，对一些较为简单的系统倒是一种简便的方法，既快而又有把握。如用 PLC 控制电梯，多年前，沈阳的一些电梯厂家多用 OMRON 公司的 P 型机，如控制 6 层楼多用 C60P，一个主机就够。这样，如果再有类似的电梯系统用它就可以了。

当然，一些复杂的系统，由于共同性少一些，不大好类比。但类比也仍可作粗略参考，特别可用以确定用哪个厂商的 PLC、什么型号的 PLC。

要注意的是，在类比时一定要考虑 PLC 的发展，如上述的 OMRON P 型机，现已不复存在了，应用 CPM2A 型机代替。

（2）估算法

估算法主要用于计算 I/O 点数，以粗略确定 PLC 的型别，用大型机、中型机、小型机还是微型机。

估算法首先要计算所需的 I/O 点数。I 点数 N_i 为

$$N_i = I(P_i - 1)E_i$$

式中 E_i——系统所使用的某类输入器件的总数，如用了 5 个按钮，则为 5；

P_i——该类器件可能处于的工作状态，如按钮，一般处于按下与松开两种状态，再如多位开关，可处于多种状态；

I——输入器件总数。

O 点数 N_O 为

$$N_O = O(P_O - 1)E_O$$

式中 E_O——所使用的某类输出器件总数；如用了一台正反转电动机即为 1；

P_O——该类输出器件可能处于的工作状态，如能正反转电动机则有 3 种状态，即正转、反转及停止；

O——输出器件的总数。

开关量总数 $N = N_i + N_O$

另外，模拟量也要估算。有多少监视量就有多少路模拟量输入；有多少控制输出，就有多少路模拟量输出。它们比较好算。

估算出 I/O 点数及模拟量路数后，可依大、中、小型机划分的大致标准，估算要选用的 PLC 机型。

一般大、中、小机型间点数均有交叉，没有把握不妨都作考虑，再用计算法进一步确定。

（3）计算法

它是估算的再精确一步。要确定用什么样模块，用多少模块，一般要作 4 个方面计算：

1）模块数计算。确定了 I/O 点数，还要按 I/O 的物理要求，确定各用什么样模块。

对输入点，要依输入信号电压区分，是交流的，还是直流的？信号间有什么隔离要求？进而确定选用多少种、各有多少输入点的模块。最后，再依总数具体计算应采用多少种，各多少个模块。

对输出点，要考虑输出形式，是继电器，还是晶闸管或其它半导体器件？还有就是公共回路，一般点数多的模块，其公共回路就少；反之则多。如 OMRON 公司的 C200H 输出模块，若为 8 点输出，则每个输出回路都是独立的。这便于电路配线，但要用的模块多。而 16 个输出点的，则 16 路输出均用一个公共回路。这样模块各输出点只能用相同的电压，否则无法工作。选定了模块类型，再依总数计算模块数。

模拟量也有相应的模块要选择。如有的是 4 路输入或 8 路输入，有的是 2 路输出或 1 路

输出，有的既有入，还有出，等等。选了合适的模块之后，再根据总的路数，计算模块数。

I/O 模块数计算之后，再算要使用的机架槽位数，进而确定机架数。

2）电源容量计算。对 PLC 电源容量也要作计算。箱体式 PLC 电源容量一般都可自动满足，可不考虑。只是在确定隔离变压器（市电电源经隔离变压器再加载到 PLC，以减少电网对 PLC 的干扰）时，对变压器的容量要稍作计算。模块式 PLC 的电源种类较多，要作相应计算，然后再选型。

3）响应时间计算。信号响应时间与 PLC 的循环时间有关，而循环时间与 PLC 的程序量有关。要计算，不太容易。不过，有的 PLC 厂商提供了经验公式，大体的关系是内存总量与要使用的点数有关，这样既可用以确定选多大容量的内存，也可大体确定循环时间。算出循环时间，就可进一步计算响应时间了。

特殊输入量的响应时间更要分别计算，有的可查有关模块的特性。

4）投入费用计算。在模块种类及数量确定后，一般还要依报价进行投入费用的计算。如果配置的方案多，对每个方案的投入费用也都要计算。精确计算费用有利于依经济性原则对系统进行配置。

（4）测试法

系统配置时，一些重要的数据不仅要计算，有的还要进行实际测试。如循环时间，可把编完的程序送入 PLC 进行实际测定。有的数据可由厂商提供，或委托厂商测试。

3. 系统配置的步骤

系统配置的步骤为

（1）用类比法，大致确定可选用的厂商产品及机型。

（2）用估算法，估算 I/O 点数及模拟量路数，并确定要选用的机型。

（3）用计算法，依完整性原则计算所需的模块数。这里可能有多个方案，那就应计算出各个方案的结果。

（4）再依可靠性原则，考虑必要的冷备份、热备份或冗余配置。若为一般系统，这个步骤可省略。

（5）计算各个方案的投入费用，并依经济性原则选其中最优者。

（6）必要时再进一步作性能计算或进行实物测试。再根据计算或测试结果，对原有的配置作修正。

4. 系统配置类型

（1）基本配置：这种配置控制规模小，所用的模块也少。对箱体式 PLC，则仅用一个 CPU 箱体。CPU 箱体含有电源、内装 CPU 板、I/O 板及接线器、显示面板、内存块等，是一台完整的 PLC，送入程序，通电后即可工作。

CPU 箱体依 CPU 性能分成若干型号，并依 I/O 点数，在型号下又有若干规格。基本配置就是选择一种合适的 CPU 箱体，来满足实际的要求。自然，PLC 厂商箱体的型号及规格多，分布得越合适，也就越便于进行这种配置。

对模块式 PLC，基本配置选择的项目要多些，有：

CPU 模块：它确定了可进行控制的规模、工作速度、内存容量等，选得合适与否至关重要，是系统配置中首先要进行的。

内存模块：它可在 CPU 规定的范围内选择，以满足存储用户程序的容量及其它性能

要求。

电源模块：有的 PLC，它是与 CPU 模块合二而一的，有的是分开的。但这两者选择的原则都相同，都是依 PLC 用的工作电源种类、规格和要否为 I/O 模块提供工作及信号电源，以及容量需要作选择。电源模块多与其它模块相配套的，型号与规格不多，容易选择。

I/O 模块：依 I/O 点数确定模块规格及数量。I/O 模块数量可多可少，但其最大数受 CPU 所能管理的基本配置能力的限制。

底板或机架：基本配置时只使用一个底板或机架，这种配置简单，原因就在于此。但底板也有不同规格，所以还要依 I/O 模块数作不同选择。有的 PLC，如 OMRON 公司的 CJ1 机，无底板，那样的 PLC 就没有什么底板或机架可选择了。

图 1-22 所示为 CJ1 机的基本配置。

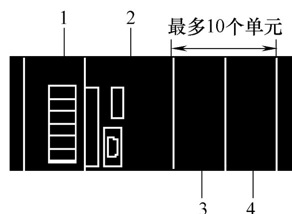


图 1-22　CJ1 机的基本配置

1—电源　2—CPU　3、4—I/O 单元

基本配置具有以下 3 个特点：

1）基本配置最简单。基本配置的 PLC，其组成部分是不能再少了。少了就有损于 PLC 的完整性，就不成其为 PLC 了。

2）基本配置最便于扩展。基本配置最为简单，所以很容易扩展，而且扩展的余地也最大。

可加扩展单元，使 PLC 能控制更多的 I/O 点；

可加特殊单元，使 PLC 能进行一些特殊的控制；

可加远程单元，使 PLC 实现远程控制；

可加联网单元，使 PLC 实现联网，等等。

3）如选用模块式的 PLC，单点的费用较高

单点费用是指用 PLC 实现控制时，每一 I/O 点分担的硬件费用。单点费用往往用它评价 PLC 配置的经济性。这个费用当然越低越好。

（2）扩展配置：箱体式 PLC 扩展配置是增加 I/O 箱体。I/O 箱体有不同的型号和规格。可按所需增加的点数，选用相应的 I/O 箱体。

模块式 PLC 的扩展配置有两种：一为当地扩展，另一为远程扩展。

当地扩展配置：在基本配置的基础上，于当地增加 I/O 模块及相应底板或机架，这可使 PLC 的控制规模有较可观的扩大；

远程扩展配置：这种配置所增加的机架可远离当地，近的几百米，远的可达数千米。远程配置可简化系统配线，而且也可扩大控制规模。

扩展配置时应着重弄清两个问题：

1）该型号 PLC 最大可能的扩展配置。显然，充分利用最大可能的扩展配置，可降低 PLC 单点费用，提高使用 PLC 的经济性。

2）I/O 地址的分配规律。显然，只有弄清了 I/O 地址是怎么分配的，才可能编写 PLC 程序。

扩展配置是在基本配置的基础上，增加扩展机架或扩展箱体，从而增加 I/O 点数的配置。

扩展配置可充分利用 CPU、内存、外设资源，使 PLC 的单点费用降低。

扩展配置还可实现远程安装，简化接线，便于维护，可降低安装与使用费用。

扩展配置依 PLC 类型的不同而有所不同，差别较大。

图 1-23 就是 CJ1 机的一个当地扩展配置的例子。

（3）特殊配置：特殊配置是指增加特殊 I/O 模块的配置。特殊配置后的 PLC 可进行哪些方面的控制，与所配置的特殊单元紧密相关。目前，各 PLC 厂商已开发或正在开发越来越多的特殊单元，使 PLC 的控制能力越来越强。

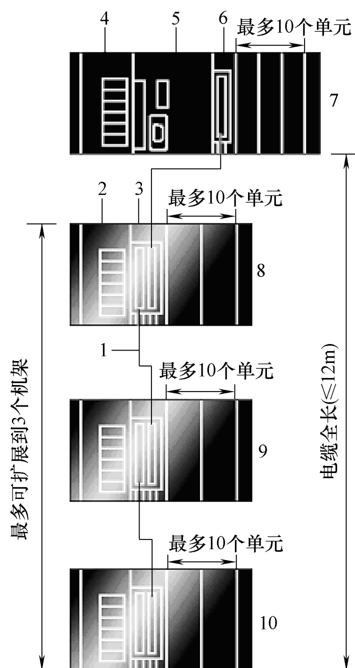


图 1-23 当地扩展配置

1—I/O 连接电缆 2—I/O 接口单元 3、4—电源 5—CPU 6—I/O 控制单元 7—CPU 机架 8～10 扩展机架

较常见的特殊模块，若按功能划分，有：

- 高速计数单元：可计入与处理高频脉冲信号，从处理几万赫，几十万赫的都有。如果与旋转编码器配合使用，可测量与处理转动或移动信号。
- 位置控制单元：可通过输出脉冲控制位置移动量及位置移动速度。有单坐标控制的，还有双坐标控制的，以至于多坐标控制的。双坐标或多坐标的还可实现两坐标运动协调。这实际是数控技术通过 PLC 特殊单元予以实现。
- 模拟量输入、输出单元：可把传感器模拟量转换成数字量存于输入通道中，或把输出通道的内容转换成模拟量作为控制信号，可对一路信号作转换，也可对多路信号作转换，可多达 8 路，以至于更多，即一个单元可处理 8 路模拟量。有的既有模拟输入、又有模拟输出，等等。
- 温度检测控制单元：温度是工业中常见的模拟量。为检测与控制它，PLC 厂商开发专门针对它的检测与控制单元。实质上它也是模拟量，只是对它作了更多的预处理工作，如可适用于不同的温度传感器，可测的温度范围也可相应选择，用起来更方便。检测单元只能用于读温度值，并可显示，也可用作处理，但要另编程序。而控制单元，可自成控制回路，无须编程，也无须与 PLC 运行相关的程序即可实现温度控制。
- 其它特殊单元：如计算机单元，可把有关个人计算机的主板、外设作成 PLC 的单元，插在底板上，可直接与 PLC 通信。再如 ASCII 单元，也具有运算与通信功能，可协助 PLC 的 CPU 作数据处理，等等。

（4）冗余配置

冗余配置指的是除所需的模块之外，还附加有多余模块的配置。目的是提高系统的可靠性。能否进行冗余配置，可进行什么样的冗余配置，代表着一种 PLC 适应特殊需要的能力，是高性能 PLC 的一个体现。一般的 PLC 是不具备这种性能的。

1）冷冗余。在配置系统后进行采购时，对易出故障的模块，多购一套或若干套作为备份，以备一旦现役的模块坏了，能及时更换，以减少故障后系统修复的时间，减少停工损失。之所以叫冷备份是因为备份的模块没有上生产线，只是放在备份库房中待用。

冷备份要备什么、备多少是颇有讲究的。过去有的工厂进口一些设备，缺乏备份，出了

问题后，临时进口备件，一拖就是几个月，耽误不少时间，其损失是相当惨重的。

但也有的冷备份数量太多，无关紧要的也备份，很不合算。特别是 PLC 技术发展很快，旧产品常被新产品所更新，备份过多，不如用新的予以取代。

2）热冗余。热冗余是冗余的模块也上线，只是不参与控制。一旦参与控制的模块出现故障，它可自动接替其工作，系统可不受停机损失。

OMRON C2000H 机、CVM1D 机及 CS1D 机的 CPU 模块就可作这种备份。图 1-24 所示为 CS1D 机冗余配置时 CPU 机架的配置。它有两个 CPU、两个电源、两个 Control Link 单元，一个工作，一个热备。

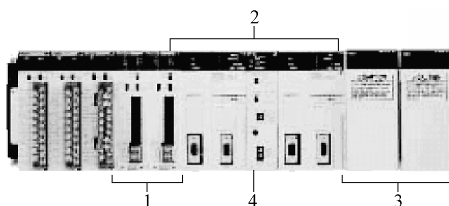


图 1-24　冗余配置
1—两个 CLK 模块（其中一个备份）　2—两个 CPU 模块（其中一个备份）　3—两个电源模块（其中一个备份）　4—双机单元

当某个 CPU 单元出现问题时，双机单元立即将控制切换到另一个 CPU 单元。能以最小的影响，使系统继续运行。

配备两个电源，可保证即使一个出现问题，而另一个仍可供电。

配备两个 Control Link 通信单元网，双回路工作。如图 1-25 所示，一个不能工作，另一个仍可维持通信。图 1-26 所示的以太网也是冗余的，一个通路断开，另一个还可工作，并能自动选择路径。

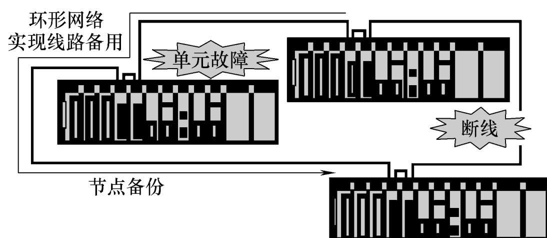


图 1-25　Control Link 网双回路工作

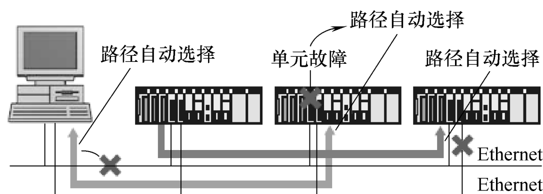


图 1-26　以太网双回路工作

3）容错冗余。在特别或非常重要的场合，为做到万无一失，还有配置成容错冗余系统的。多套 PLC，如 3 套 PLC，同时工作，其输出依少数服从多数的原则裁决。这种系统，即使有少数 PLC 出现故障，系统仍可正常工作。当然，这种容错冗余系统也只是重要的场合才这么做。

冗余配置的目的是提高整个系统的可靠性。冗余配置为何能提高可靠性？这是因为：PLC 是由各种模块组成的，而各模块都有可靠度。在 PLC 非冗余配置时，任一模块出现故障，都会影响 PLC 工作。按若干独立事件构成的事件的概率计算原则，PLC 的可靠度应为组成它的各模块可靠度的乘积，即

$$R = R_1 R_2 R_3 \cdots R_n$$

式中　R——PLC 的可靠度；

$R_1 \sim R_n$——各模块的可靠度，指在规定使用寿命内可靠工作的概率。用 1 减去它，即模块的失效率。

如各模块的失效率为万分之一，则其可靠度为

$$1-1/10000=9999/10000$$

如 PLC 由 10 个模块组成，则可靠度近似为

$$(9999/10000)^{10}\approx 999/1000$$

其失效率为 $1-999/1000=1/1000$

若为 100 个模块构成的系统，其失效率将为 1/100。可知，系统越大，组成的模块越多，出现故障的概率也越大。

但若为冗余配置，情况就不同了。热备份也好，表决系统也好，必须至少有两个模块出现故障，系统才不能工作。这两个模块同时出现故障这种独立事件，构成系统出故障的合成事件，其概率关系也是乘，只是失效率的乘。同样用以上例子作讨论，若这 100 个模块均为热备，或三选一，则只有在两个相同的模块同时出现故障时，系统才失效。其故障率的计算可先分别算出各模块的失效率，再合成算 PLC 的可靠度，即

由于各模块为热备份或三选一，两个同时失效，模块才失效。两个模块同时失效的概率为（同样以单模块失效率为万分之一计）：

$$(1/10000)\times(1/10000)=1/10^8$$

即亿分之一的几率。

再算 PLC 失效率的概率为

$$1-(9999999/10^8)^{100}\approx 1/1000000$$

即为百万分之一，确实是万无一失。

当然，实际上不会所有的模块都三取一或热备，所以失效率会比这个大。

（5）联网配置

为提高控制性能，往往要把在地理上处于不同位置的 PLC 与 PLC，或 PLC 与计算机，或 PLC 与其它控制装置或智能装置通过传送介质连接起来，实现通信，以构成功能更强、性能更好的控制系统。

图 1-27 所示就是 OMRON PLC 联网的概况。

为了联网，PLC 就要有与其相应的网络模块。要不要联网，如何联网，选用什么样的网络模块，也是在配置 PLC 时要考虑的一个重要侧面。

（6）附加配置

附加配置是为 PLC 配备外部设备所作配置。其目的是为 PLC 程序的编制、调试、存储，以及数据的显示、存储及打印提供条件。

1.6.2 程序编制

编制 PLC 程序要按步骤进行。其步骤为

1. 弄清工艺

弄清工艺，首先要弄清使用 PLC 的目的，要用到 PLC 的哪些功能。其次，要弄清两方面情况：一为输入、输出部件的特性与分布，即系统的空间情况；另一为系统工艺过程，即系统的进程情况。

（1）空间情况。弄清各输入部件的性能、特点，并分配相应的输入点与其连接。分配时，既要考虑布线简单，还要避免信号受外界干扰。弄清各输出部件的性能、特点，分配相应的输出点与其接线。如可能，接输出部件的模块最好能与输入部件的模块能适当隔开，以

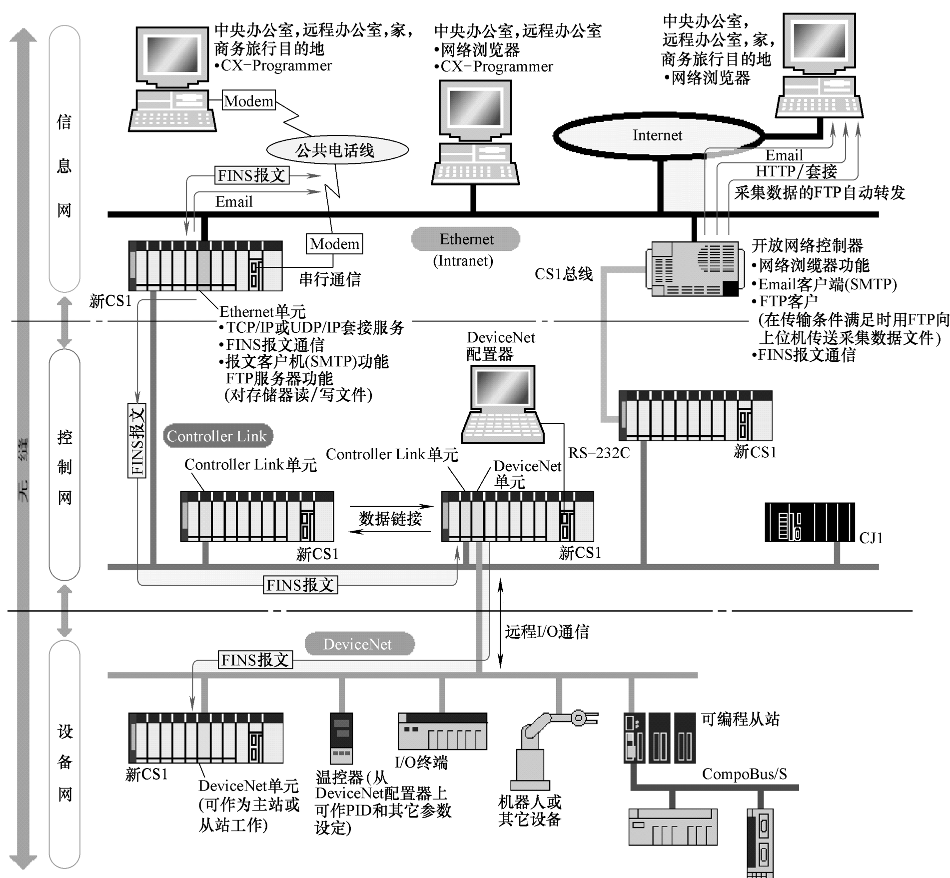


图 1-27　OMRON PLC 联网的概况

避免输出信号对输入的干扰。此外，还要考虑在编程时地址使用的方便。弄清了这些，才便于合理地分配 I/O 地址。

（2）进程情况。弄清被控对象的工作要求、工艺过程及各种关系。弄清其工艺过程，看它是怎么开始的？怎么展开的？怎么终止的？弄清输出与输入的对应关系，如果存在时序关系时，两者的时序是怎么对应的？弄清要采集、存储、传送哪些数据？弄清有哪些互锁、联锁关系？有那些特殊要求？等等。弄清这些问题才能着手设计算法，也才能进一步进行程序设计。

2. 硬件设定

为了 PLC 能按要求工作，在使用 PLC 之前，要对 PLC 的硬件作必要的设定。如设定特殊模块的机号、设定扩展指令功能号、PLC 上电时工作模式（是运行、监控，还是编程）等。有的厂生产的 PLC 还可对 PLC 的内部器件（如要使用多少定时器、计数器等）进行分配或指定。

PLC 出厂时，厂商多有其默认设定，但对较复杂的系统，用户必须有合乎自己情况的设定。一般说，硬件设定在开始编程之前是必须进行的。

3. 分配 I/O

PLC I/O 点在模块或在箱体上的地址是固定的，在模块或箱体上都有相应的地址标记。

而模块与箱体上的地址（通道号）是按一定规律分配的。只是不同厂商、不同型号的 PLC，有不同的规律。常见的规律有：固定分配、定位分配、顺序分配及设定分配。

固定分配是固定地分配通道地址。如 OMRON 公司的小型机就是固定分配，其主机的通道地址与主机的点数有关。扩展箱体的通道地址依远离 CPU 箱体升幂而依次递增。

定位分配是按模块所在的机架及其在机架上的位置分配其通道地址。模块位置定了，其通道地址也就确定了。

顺序分配是按模块在 PLC 中位置顺序，依次升幂分配通道地址。依模块点数的不同，有的占 1 个通道（点数不足 1 个通道的，按 1 个通道计算），有的占 2 个通道，甚至于更多。OMRON 公司的大型机就是这么编号的，OMRON 公司的 CQM1 机也是这么编号，只是它把 I 与 O 分开，分别按其顺序进行排列。

设定、分配是在指定的范围内，通过硬件或软件设定、分配模块通道地址。OMRON 公司的特殊模块，其所用的通道地址就是靠指定机号后设定的。CJ1 机可设定机架地址。

由于当今编程软件的进步，一般都可用符号地址编程。这就有可能先用符号地址编程，编好后，再编辑符号与实际地址的对应关联。这么做时，编写程序可先做，而硬件设定、I/O分配后进行。但程序下载、调试前，这些工作都必须做好。

4. 编写程序

首先是考虑程序的组织，可按功能，把程序先划分成若干模块。分模块编程，然后再予以合成。按模块编程便于移植一些已用过的程序，而且也便于调试。

其次，分块设计算法。算法确定后，其思路可用框图或一些自然语言表达。算法在对工艺进程的分析中形成，是编写程序的基础与准备。

最后，按模块逐一编写指令。要逐条编写指令，若为梯形图编程，则要逐个画出图形符号，最终要形成一个指令集，或完整的梯形图。

5. 调试程序

编写 PLC 程序是很细致的工作，差错总是难免的。而任何一点差错，即使是一小点，都可能导致 PLC 工作出现故障，所以编写程序后，还要进行调试，纠正种种差错。

调试程序可通过计算机仿真进行。多数公司现在都有相应的仿真软件，可运行在这软件平台上对所编的程序作仿真调试。

多数的程序调试是把程序送入 PLC，在 PLC 试运行（输入、输出不接传感器及执行机构）时作调试。这也叫在线调试。

在线调试可使用简易编程器，先把程序送入 PLC，然后分模块或分指令一步步地调试。

在线调试也可使用计算机，由相应软件协助进行：先把程序录入计算机，再下载到 PLC；然后使 PLC 运行，通过计算机画面了解 PLC 运行情况，观察其是否与设计意图符合；不符合，则找出原因；再修改程序，剔除毛病；再试，再看，再找，再改，一直到合乎设计意图。

经在线调试的程序，还要在现场联机调试。只有经联机调试合乎要求的程序，才是合格的、可交付用户使用的程序。

6. 存储程序

把程序录入计算机后，就要作存储，甚至开始编程时，编一部分就要存储一部分。随着程序调试通过及试运行过程的不断完善，还要不时地存储。存储时，一般只留下后来的，删

除过去的。程序不仅存于 PLC 的 RAM 中，也可存入磁盘或磁带中。

经试运行后的程序可作定型。办法是把它固化，写入 ROM 中。

程序保护：

硬件：有的 PLC 用硬件开关设置程序保护。读写 DIP 开关 ON 保护，否则不保护。

软件：有的用软件设定保护，如 CPM 机是 DM6602 字的 0 位，设为 1 保护，0 不保护。

程序加密：程序加密可保证程序不被删除或修改。但其它人可读它，重用它。为了保护知识产权，可对程序加密。

PLC 程序加密的方法有：用指令加密；用编程软件加密；可全程序加密，也可局部加密。

程序加锁：除了程序保护、加密；还可对程序加锁。可做到即使 PLC 程序正常运行，但不产生控制输出。加锁可用置位 PLC 的输出禁止位实现，也可用自编一段小程序，使相应的输出禁止。

这里，只是对 PLC 编程作一最简略的说明。详细的介绍将是本书的主要任务。以下各个章节，将对 PLC 编程的方方面面问题，逐一进行详细讨论。

结语

基于计算机技术的 PLC，其工作基础是信息的获得、变换与发送。信息具有便于处理、传递、存储、可反复重用等特点。再加上 PLC 具有可靠的 I/O 系统及种种接口，给 PLC 带来了无限的生命力。也决定了 PLC 非常有用、好用、耐用及省用。PLC 成为当今系统自动化、信息化、远程化及智能化的重要支柱，已是顺理成章之事。

请想想

1. PLC 的实质是什么？
2. 信息有哪些特点？试设想用一个开关通过 PLC 去控制一盏灯，该怎么实现？
3. PLC 主要功能有哪些？
4. 可从哪些特点划分 PLC？
5. PLC 有哪些基本特点？
6. 用好 PLC，要在软、硬件方面做好哪些工作？

第 2 章　PLC 编程技术基础

PLC 编程技术是有关编写、调试 PLC（用户）程序的技术。掌握好这个技术，才能正确与有效地使用 PLC。

本章将介绍 PLC 编程技术的基础知识。

2.1　PLC 编程语言

关键词：编程语言、指令、程序、助记符、梯形图、IEC、IEC 61131 国际标准、GB/T〖5969、结构化文本语言、功能块图、连续功能图、顺序流程图、系统流程语言（SYSFLOW）、SAMA 图、数控用 G 语言。

PLC 程序是 PLC 指令有序集合。而指令（Instruction）也有的厂家叫操作（Operation），是用以告知 PLC 作什么、怎样去作的文字代码或图形符号及其使用的操作数。所以，PLC 程序也可认为是文字代码、图形符号加上相关数据的有序集合。

从本质上讲，指令都只是一些二进制代码，即机器码。PLC 编程器或 PLC 编程软件可把用不同的编程语言编写的程序编译成机器代码。所以，用户所看到或用到的 PLC 指令一般不是机器代码，而是文字代码，或图形符号。

传统 PLC 编程语言只有两种，指令表（Instruction List，IL。有的称布尔助记符，Boolean Mnemonic）及梯形图（Ladder Diagram，LD）。

而今，PLC 在各个自动化领域应用的不断推进，为便于各类型的工程技术人员都能使用 PLC，PLC 厂家都增加了它的编程语言。国际电工组织也制定与几次修订了 PLC 编程语言国际标准。1993 年，作了全面修订后称之为 IEC 61131-3 的修订版。我国在 1995 年 11 月发布了 GB/T 15969-1/2/3/4 标准，与 IEC 61131-1/2/3/4 等同。

IEC 61131-3 标准推荐了 6 种编程语言。除了指令表及梯形图，还有结构化文本（Structured Text，ST）、功能块图（Function Block Diagram，FBD）、连续功能图（Continuous Function Chart，CFC）及顺序功能图（Sequential Function Chart，SFC）。

所推荐的编程语言有如下优点：

开放性，能最大限度地运行于不同制造商的 PLC。

灵活性，一个程序的不同部分可使用不同语言。

先进性，支持结构化程序开发、复杂的过程控制及结构化数据。

可靠性，有很强的错误检查和纠正能力。

等等。

目前 IEC 61131-3 编程语言，不仅用于 PLC，而且还用于集散型控制系统、工业控制计算机、数控系统和远程终端单元。

然而，由于这个标准的建立是在 PLC 已广泛使用之后，加上它不是强制性标准，所以，有些老的 PLC 厂家多还是在原来语言的基础上，作了扩展。并没有完全采用这个标准。如

日产 PLC 多数就没有采用功能块图语言。再就是，即使语言名称相同，但细节还是有不少差异。

倒是我们国产 PLC，如和利时公司的 LM、LK 系列机，是在有了标准之后才开发的。没有与原有 PLC 的兼容问题，所以能全面采用这个标准。该机可使用标准规定的 6 种语言编程。这也许是后来居上吧！

以下对这 6 种语言分别作简要介绍。

2.1.1　指令表

指令表又称助记符、列表，是基于字母符号的一种语言，类似计算机的汇编语言，用拼音文字（可用多国文字）的缩写及数字代表各相应指令。这种语言在欧洲很常用。绝大多数 PLC 都使用有这种语言。如下程序就是由指令表语言编写的：

```
0′这个 0 为一个条的开始。条是由含义完整的输入（前因）到输出（后果）变换的若干指令组成。本例共 2 个条。0 条有 4 个指令，1 条仅一个指令。
    0  LD  0.00（*装载指令*）
    1  OR  10.00（*或运算指令 LM 机为 OR  %QX0.0，位地址用% 号标识*）
    2  AND  NOT  0.01（*取非运算之后再做与运算指令*）
    3  OUT  10.00（*输出指令*）
1   4  END（*表示程序结束*）
```

上述每个“（*”与“*）”之间为 IEC 61131-3 标准的指令注解。每个指令都可加注解。上述程序中符号“′”开始的文字为 OMRON PLC 指令注解。但每条只能加注一次，而且只能在“条”的开始处加注。

这里共用了 2 个条，共 5 个指令。除第 1 条中的第 4 个指令外，其它都含有 3 个部分：

（1）指令地址。标志该指令存于 PLC 程序存储区的位置。一般讲，指令总是从 0 地址的指令开始顺序执行，一直执行到最后一条指令为止。但由于编程工具及编程软件的发达，在送入指令时，这个地址多是自动生成的。所以，有的 PLC，如 LM 机就不用它。

（2）操作码。用以告知 PLC 应该进行什么操作，是 PLC 指令的核心，是必不可缺的。

（3）操作数。是操作码操作的对象。有一个操作数的，两个操作数的以及多个操作数的。也有无操作数的，如这里的 END 指令，它只是表示程序到此结束。到底有多少操作数视操作码而定。

指令表容易记忆、便于操作。与其它语言还都可一一对应。而且，其它语言无法表达的，用它都可表达。是 PLC 最基本的编程语言，也是简易编程器使用的惟一编程语言。

2.1.2　梯形图

它源自继电器电气原理图，是一种基于梯级的图形符号布尔语言。它通过连线，把 PLC 指令的梯形图符号连接在一起，以表达指令调用及其前后顺序关系。

梯形图的连线有两种：一种为母线，也称电源线，画在梯形图两边，左方称左母线，右方的称右母线，用于梯形图指令间的整体连接；另一种为内部小横线与小竖线，用于梯形图指令间的局部连接。

有了母线，可把图形连接成连通的整体（但，有的厂家母线不是连通的）。为了方便，

右母线可省略。这样的图形类似于梯子，梯形图因此而得名。

有了内部横、竖线，可把若干个梯形图符号（即指令）连成一个含义明确的图形符号组合，即上述“条”，有的厂家称之为梯级（Rung，有的称为 Nextware）。它是一组前后连贯，能代表一个完整的逻辑含义的梯形图符号集。是设计梯形图程序的最基本单位。

与原有的继电器逻辑控制技术不同的是，梯形图中的两电源线不是实际意义的电源，内部的继电器也不是实际存在的继电器，因此，应用时，需与原有继电器逻辑控制技术的有关概念相区别。

提示：梯形图的左右母线好像电气原理图的电源线一样，一般不直接与输出类指令（相当于电气原理图的负载）相连，中间要有一些建立逻辑条件的指令（相当于电气原理图的控制元件）。但有的厂家也允许这么做。

图 2-1 所示为 OMRON 公司用的梯形图，该图未标明操作数。

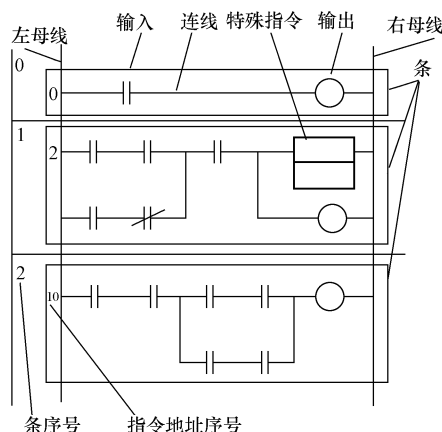


图 2-1 OMRON 公司用梯形图

该图共有 3 个“条”。从图知，这里条序号是连续的。而指令地址序号则不连续。因为条 0 有 2 个指令，要占 2 个地址。所以，条 1 开始的指令地址应为 2。同理，条 2 开始的指令地址应为 10。尽管条 1 只有 7 个图形符号，但它的图形关系复杂，有一个特殊指令，故 7 个图形符号用了 8 个地址。

梯形图中的指令地址序号代表指令执行的顺序。梯形图指令地址顺序一般为，先上后下，先左后右。即图上方、左方的梯形图指令先执行，而下方、右方的梯形图指令后执行。而且，这个顺序（含“条”的序号），都是在编程时，由编程软件自动生成的。为了简化，本书用的梯形图程序，有时把条及指令地址序号省略。

梯形图符号编的 PLC 程序，很像电气原理图，较易为电气工作人员理解。联机调试、观察 PLC 操作状态时，非常生动、直观。图 2-2 所示即为图 1-14 监控时工作的情况。它表示 10.00 点已工作。

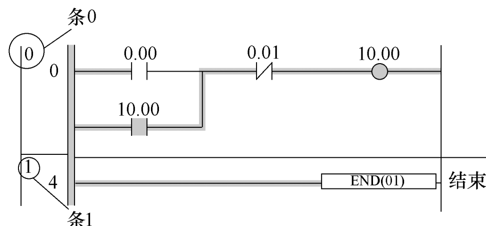


图 2-2 在线监控梯形图

正是梯形图语言有诸多优点，所以，它已成为 PLC 编程的基本语言，特别在北美已经得到广泛应用。用它表达的顺序关系，不如用助记符表达得清楚，弄不好，易出现二义。图形过分复杂时，还容易出错。所以，有的 PLC 程序如不能用梯形图表达时，最终还是要用助记符表达。此外，用梯形图指令编程，须使用个人计算机及编程软件。同时，PLC 还要配置通信接口。

正是梯形图语言用得很多，所以，本书介绍程序实例时，主要是用这种语言。

2.1.3 功能块图

PLC 还用有功能块（FBD）语言。它是一种对应于逻辑电路的图形语言。FBD 广泛地

用于过程控制。功能块有输入端、输出端。如图 2-3 所示为国产 LM 机用的功能块程序。这里有两个功能块：一为逻辑“OR”功能块；另一为“AND”功能块。前者的输出作为后者的一个输入。

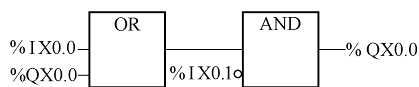


图 2-3　功能块图

该图的“OR”块类似于逻辑电路的“或门”，含义为逻辑或。“AND”块类似于逻辑电路的“与门”，含义为逻辑与。该图“OR”块有两个输入，一个为它对% IX0. 0，另一为% QX0. 0。“OR”的输出直送给“AND”块。“AND”块的两个输入一来自“OR”块，另一来自% IX0. 1 的非（这里的小圆圈为逻辑非之意）。AND 功能块的输出为% QX0. 0。显然，图 2-3 的表达的操作数间逻辑关系，与本书第 1 章的图 1-14 程序相似，也是起、保、停逻辑。

功能块语言是用图形化的方法，以功能模块为单位，描述控制功能。逻辑关系清晰，便于理解。特别是控制规模较大、控制关系较复杂的系统，用它表达将更为方便。

此外，一些含有标准功能的程序，用功能块语言则更便于调用。目前，PLC 厂家在推出一些高功能及高性能的硬件模块的同时，多提供与其有关的功能块程序，这为用户使用这些硬件模块及进行编程提供了很大方便。

功能模块语言占用内存较多，执行时间也较长，因此，这种设计语言多只在在大、中型可编程序控制器和集散控制系统的编程和组态中才被采用。OMRON PLC 即使为大、中型机也不用这种语言。但提供有供梯形图语言使用的系统功能块。

2. 1. 4　连续功能图

它与功能块语言类似。也是按需要选用种种功能块。每个功能块也有输入、输出。块间的联系也用连线。所不同的是，它更灵活，块的位置可任意摆放，特别有信号反馈时，画起来更方便。

为了块的执行有明确的顺序，它的每个块的右上角都标有序号。而这个标注，在功能图语言中是没有的。当然，实际表达时，也可选择不显示这个标号。

图 2-4 所示为国产 LM 机用的连续功能图程序，也是起、保、停逻辑。这里的功能块就是随意摆放的。它的输出% QX0. 0 以反馈的形式作为功能块 OR 的输入，但它有标号。所以，肯定是先“或”、后“与”，再输出，不会有二义。

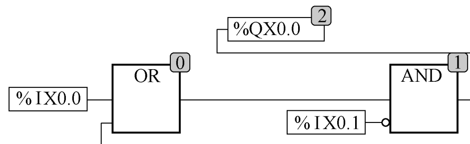


图 2-4　连续功能图程序

提示：功能块图及连续功能图语言在 DCS 系统编程中用得较多。此外，由于这两种语言差别不大，有时，仅使用功能块图语言。因而，有的也把 IEC 61131-3 自动化编程语言说成 5 种，而不是 6 种。

2. 1. 5　结构化文本语言

结构化文本语言是基于文本的高级编程语言。它采用一些描述语句，来描述系统中各种变量之间的各种关系，以执行所需的操作。它与 BASIC 语言、PASCAL 语言或 C 语言等高级语言相类似。但为了应用方便，在语句的表达方法及语句的种类等方面都进行了简化。以下

就是用 ST 语言的赋值语句。它把一组变量进行逻辑运算，然后再赋值给变量“work”。

```
work：=（start or work）and not stop;(＊赋值语句 ＊)
```

这里“work”、“start”及“stop”为布尔变量。使用之前一般要先定义。在“(＊”与“＊)”之间为程序注解。

此外，也可用条件语句实现与上述相同的功能。此语句为

```
IF stop THEN
  work：=FALSE;(＊如果“stop”为真，“work”为假＊)
ELSE(＊否则，即“stop”为假＊)
  IF start or work THEN (＊这时，如果“start”或“work”为真，则“work”为真＊)
    work：=TRUE;
   END_IF;
END_IF;
```

它表达的也是起、保、停逻辑。

结构化文本语言功能比图形语言强，可读性比指令表语言好。用它编写复杂的程序，既方便，又易读，是很有发展前途的 PLC 编程语言。它也是计算机编程人员的偏爱。但是，结构化文本语言对编程人员的计算机的技能要求较高，而且不如图形语言直观。所以，目前用的还不大普及。OMRON PLC 只是在自编功能块的程序中，可使用它。本书介绍的程序，部分也将使用这种语言。

2.1.6 顺序功能图

顺序功能图语言采用顺序功能图的描述程序结构，把程序分成若干“步”（Step，S），每个步可执行若干“动作”。而“步”间的转换靠其间的“转移”（Trans，T）的条件实现。至于在“步”中要做什么，在“转移”中有哪些逻辑条件，则可使用其它任何一种语言编程实现。

功能图来源于佩特利（Petri）网，由于它具有图形表达方式，能较简单和清楚地描述并发系统和复杂系统的所有现象，并能对系统中存有的像死锁、不安全等反常现象进行分析和建模，并可在此基础上直接编程，所以，得到了广泛的应用。

其实，顺序功能图语言仅仅是一种组织控制程序的图形化方式。其使用时，要与其它语言配合，否则无法实现其功能。所以，严格地讲，它不能算是完整的编程语言。

图 2-5a 所示的是一段“顺序功能图语言”编写的程序。是国产 LEC G3 机用的语言。

图 2-5a 中 Init 为“起始步”，Step1、Step2 为“步”，Trans0、Trans1 为“转移”。“转移”的条件是位逻辑值。为 1 转换，进入下一步；为 0 不转换，停留在所在步，执行所在步的程序。图中

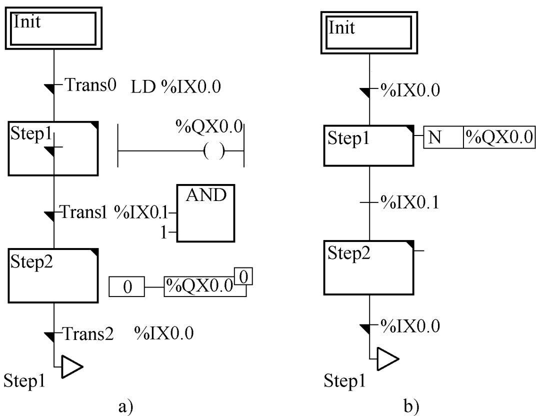


图 2-5 顺序功能图程序

a）ProwerPro 步 b）IEC 步

Trans0 条件的程序是用指令表语言编写的；Trans1 条件的程序是用功能块图语言编写的；Trans2 条件的程序是用结构化文本语言编写的。从图中可知，Step0 步转换到 Step1 的条件是% IX0. 0 ON；Step1 步转换到 Step2 的条件是% IX0. 1 ON；而 Step2 步转跳回到 Step1 则取决于变量% IX0. 0 取值，ON 则跳回。

在 Step1 中的程序是用梯形图语言编写的，而在 Step2 步的程序则是用连续功能图语言编写的。从图中可知，在 Step1 时,% QX0. 0 ON，而在 Step2 时，用 0 写% QX0. 0，也就是使% QX0. 0 OFF。因此这里表达的也是起、保、停逻辑。

图 2-5b 所示的也是一段顺序功能图语言编写的程序，但它使用了 IEC 步。是国产 LEC G3 机也可使用的与西门子 PLC 兼容的顺序功能图语言。表达得更简单些，图示程序的功能也是实现起、保、停控制。

顺序功能图编程语言的特点是：

（1）以功能为主线，条理清楚，便于理解和沟通。

（2）大型的程序，可分工设计，节省编程和调试时间。

（3）常用于规模较大、关系较复杂的系统编程。

（4）程序扫描周期短。因为，只在已“激活”的步中指令才被扫描，在未“激活”的步中的指令不扫描。

（5）只能用于程序的结构设计，真正使用必须与其它语言配合。

OMRON PLC 的 CV 系列机使用有这种语言。但后来的机型不用这种语言，而用程序的多任务组织做程序结构设计。

2. 1. 7 系统流程语言

以上介绍的 PLC 编程语言虽是国际标准语言，还是太专业了，不易被非专业人员接受。还应当有类似计算机可视化或组态软件那样的语言。OMRON 公司 C1000F、C120F 等机型用的系统流程语言也许就是这样语言。

系统流程语言所定义的指令也是告知 PLC 该做什么及怎样去做的图形符号，但不与电气硬件符号与规则相关联，也不与计算机的逻辑符号相关联。

系统流程语言只与说明系统工作的流程图（Flow chart）相关联，故可很容易被熟悉系统、但不懂电气、不懂计算机的人接受。

1. 编程过程

（1）画出过程流程图（Process Flow Chart），说明控制系统任务及可编程序控制器作用。

（2）画简化流程图（General Flow Chart），明了、简单地说明控制动作执行顺序。

（3）画详细流程图（Detailed Flow Chart），把简化流程图细化，目的是能依此编写指令。这相当于梯形图语言的梯形图。

（4）编写指令。在详细流程图的基础上，要编写指令码与数据（地址）。这个工作，相当于从梯形图到语句表的翻译工作。

2. 实例说明

用 PLC 控制机械手工作，其工作情况如图 2-6 所示。

（1）工作过程。起动→臂转到位置 A→取工件→臂转到位置 B→放工件→臂再到位置A→……如此不断重复着。

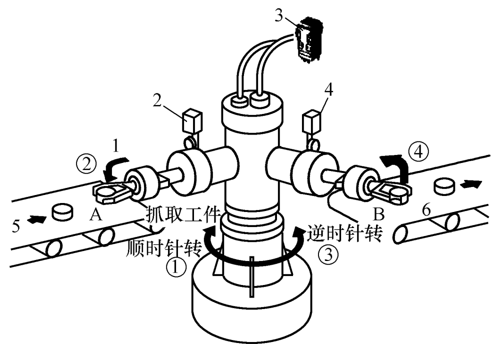


图 2-6　机械手工作概况

1—LS3 检查工件抓取　2—LS1 顺时针转　3—PB1 起动按钮

4—LS2 逆时针转　5—传送带 A　6—传送带 B

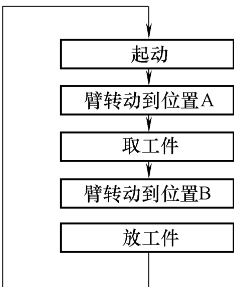


图 2-7　工作过程示意图

（2）画简化流程图。用流程图表达这个过程，如图 2-7 所示。

（3）画详细流程图。把图 2-7 的流程细化，用符号表示，标以操作数，即为详细流程图。

操作数根据动作而定。如起动，它是输入动作，其操作数即为起动按钮接到 PLC 的输入点。如点号为 0000，则起动的操作数为 0000。再如臂转动从 B 到 A，它是输出动作，若这个动作的电磁阀线圈接到 PLC 的输出点，点号为 0200，则这个动作的操作数为 0200，等等。

动作则用符号表示，输入动作这里用 AND，输出用 OUT。因为输入条件有个是否具备的问题。条件不具备，就要等待。故等待也是动作，其符号用 WAIT。

图 2-8 中未解释的还有 LBL0 为标号 0，为指令在图或程序中的位置标识。JMP 为跳转，执行它，要求 PLC 跳转去执行标号指定的指令。OUT 之后再加 NOT，表示输出动作禁止。与原梯形图的 OUT-NOT 含义相同。

此外，详细流程图还要把一些动作分解开。如臂转到位置 A，分解成向位置 A 转动及等待是否到位置 A 两步。事实上，PLC 对动作的准确控制也要有这两步。

详细流程图的一个步，用一个框表示。框中的符号为动作，数字为点号。

（4）编写指令。有了详细流程图，即可编写 PLC 的指令表。表 2-1 即为与图 2-8 对应的指令表。

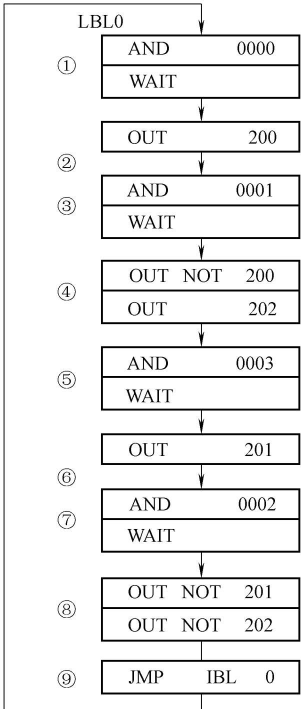


图 2-8　详细流程图

表 2-1　指令表

地址	操作码	数据	地址	操作码	数据
0000	LBL	0	0008	AND	0003
0001	AND	0000	0009	WAIT	
0002	WAIT		0010	OUT	201
0003	OUT	200	0011	AND	0002
0004	AND	0001	0012	WAIT	
0005	WAIT		0013	OUT NOT	201
0006	OUT NOT	200	0014	OUT NOT	202
0007	OUT	202	0015	JMP IBL	0

把表中的语句通过编程器（与梯形图语句用的编程器不同）送入 PLC，再运行程序，如其它条件也具备，即可使 PLC 进行上述动作的控制。只要起动条件具备，机械手将不停地从 A 处取工件放到 B 处。若起动条件不具备，则机械手停止在 B 处。

由于系统流程语言有等待指令，其执行程序不同于梯形图语言。如本程序用了多个等待，这里的第一个等待条件不满足，即 0000 不为 ON，则 PLC 就不执行后继指令，只在①处循环等待。0000 ON 后，执行②，继而执行③，并又在③处等待。它的程序不是周而复始地不断执行着，而是用到哪个指令就执行哪个指令。

3. 系统流程语言特点

（1）较容易理解。不懂得电路的人，也可理解。尽管这里介绍的只是几个最简单的指令。

（2）对输入的响应比梯形图语言快。因为梯形图语言中的指令，有用、无用都要执行，而系统流程语言则需要时才执行，总的执行指令周期短。

只是使用这种语言的 OMRON PLC，目前还很少进入中国市场，可能今后会有所变化。

2.1.8　其它编程语言

SAMA 图语言是 Scientific Apparatus Makers Association（SAMA）科学仪器制造商协会开发的一种工程组态图语言。包含各种图形符号（功能模块）和连接线（通常情况下，模拟量为实线，逻辑量为虚线）。

SAMA 图使用各种图形符号，如加、减、乘、除、微分、积分、或门、与门、切换、最大值、最小值、上限幅、下限幅等，表达控制系统的运算与控制处理。图 2-9 所示为它的常用的图形符号。



图 2-9　SAMA 图形符号

SAMA 图在 DCS 中用的很多。OMRON 的回路控制模块使用的编程语言也与此类似。集成在 CX-one 中的 CX-Process Tool 软件就是用这个语言编程。

此外，还有数控用 G（代码）语言。这在 OMRON PLC 调用运动模块时也用到它。集成在 CX-one 中的 CX-Motion 软件就是用这个语言编程。

本节已介绍了很多编程语言，但不管用什么语言编程，一般讲，同一公司的产品，几种语言的程序多数都有对应关系，其程序都可从一种语言，转换成另一种语言。所以，本书使用的主要是梯形图语言。但一些复杂的程序也使用了 ST 语言。

提示：其实，只要有相应的编译软件，什么语言，以至用自然语言，也都可用以编程。

2.2　PLC 软器件

关键词：软器件、位、数位、字节、字、通道、输入继电器、输出继电器、I/O 登记、定时器、计数器、CIO、DM 区、EM 区

2.2.1　概述

PLC 软器件是指 PLC 的 I/O 内存区（I/O Memory Area）。它所存储的数据也就是 PLC 指令的操作对象，即指令的操作数。

PLC 供用户使用的内存，除了 I/O 内存区，还有用户程序区（User Program Memory）及参数区（Parameter Area）。用户程序区用于存储用户程序。参数区用于存储系统设定参数。

不同的 PLC，这些区分布及地址也是不同的。如 OMRON 公司的 CJ1 机，其参数区、I/O内存区在 PLC 内存中的地址，以及对应的用户使用的地址见表 2-2。

表 2-2　CJ1 机内存地址分配

类　　型	PLC 内存地址(十六进制)	用户地址	区　划　分
参数区	00000 ~ 0B0FF	—	PLC 设定区 I/O 登记区 路径表区 总线模块设定区 实际 I/O 表区 模块概要区
I/O 内存区	0B100 ~ 0B1FF	—	系统留用
	0B200 ~ 0B7FF	—	系统留用
	0B800 ~ 0B801	TK00 ~ TK31	任务标志区
	0B802 ~ 0B83F	—	系统留用
	0B840 ~ 0B9FF	A000 ~ A447	只读辅助区
	0BA00 ~ 0BBFF	A448 ~ A959	读写辅助区
	0BC00 ~ 0BDFF	—	系统留用
	0BE00 ~ 0BEFF	T0000 ~ T4095	定时完成标志区
	0BF00 ~ 0BFFF	C0000 ~ C4095	计数完成标志区

续

类　　型	PLC 内存地址(十六进制)	用户地址	区　划　分
I/O 内存区	0C000～0D7FF	CIO 0000～CIO 6143	CIO 区
	0D800～0D9FF	H000～H511	保持区
	0DA00～0DDFF	—	系统留用
	0DE00～0DFFF	W000～W511	工作区
	0E000～0EFFF	T0000～T4095	定时器现值区
	0F000～0FFFF	C0000～C4095	计数器现值区
	10000～17FFF	D00000～D32767	DM 区
	18000～1FFFF	E0_00000～E0_32767	EM 区 0 段
	20000～27FFF	E1_00000～E1_32767	EM 区 1 段
	28000～2FFFF	E2_00000～E2_32767	EM 区 2 段

早期 PLC 突出的是逻辑量控制。为便于使用，PLC 的 I/O 内存区都是按功能划分，并用电器含义命名。如某某继电器、某某定时器、某某计数器，等等。其实，这些都只是内存区中的一个字或位。这些字的值或位的状态，即代表着这些器件的状态。也正是在物理上，这些器件只是内存区的字或位，所以，才称之为软器件。

1. 位（bit）、数位（digit）、字节（Byte）及字（word）或通道（channel）说明

（1）位（bit）是指二进制数的一个位，仅 1、0 两个取值。用它代表开关触点，或继电器的触点及线圈。1 代表有关开关触点通（ON）或有关继电器线圈得电（工作、ON）。0 代表这开关触点断或这继电器线圈失电（不工作、OFF）。

继电器的触点有常开的，它的线圈不工作（或说失电）时，它断、OFF；它的线圈工作（或说得电）时，它通、ON。还有常闭的，它的线圈 ON，它 OFF；它的线圈 OFF，它 ON。只是这里继电器既无实际的线圈，又无实际的触点。这里讲的线圈、触点，都只是内存单元的一个位（bit）。但在性能上可认为：线圈得电，为用 1 写了这个单元；线圈失电，为用 0 写了这个单元。用其常开触点，指直接用它的值；用其常闭触点，指用它的值取反。这样的继电器也称软继电器。显然，其触点的使用次数是不受限制的，其线圈被改写的次数也是不受限制的。这也是它比实际继电器好用的重要一点。

（2）数位（digit）由 4 个二进制位构成。这数位可以是 0（0000）～9（1001，用于 BCD，也称二十进制码），到了 9（1001）再加 1，变为 0，并进位；也可是 0（0000）～F（1111，用于十六进制数的表示），到了 F（15，1111）再加 1，变为 0，并进位。

（3）字节（Byte）由两个数位，或 8 个二进制位构成。可表达为 BCD 码，也可表达为十六进制码，还可与 ASCII 码对应。如字符 A，ASCII 码为 01000001，正好一个字节。

（4）字（word）由两个字节，或 4 个数位，或 16 个位构成。可表达为 BCD 码，也可表达为十六进制码，还可与两个 ASCII 码对应。如字符 AB，ASCII 码为 0100000101000010，正好两个字节，16 位二进制数。一个字中的字节、数位及位都有一定的编号。如图 2-10

所示。

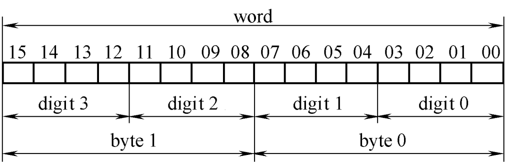


图 2-10 字组成

如果这个字与输入、输出点有对应关系，又称为通道（channel）。但有的 PLC，如施耐德 PLC 把与输入、输出点称为通道，这里意义的通道仍称为字。

使用“位”地址，除了标明它的位号，还要标明它所在的字或通道地址。如地址 0001.03，则表示为 0001 通道的 03 位，如图2-11 所示。

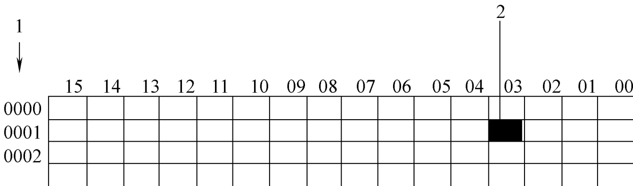


图 2-11 通道（字）地址与位地址

1—通道（字）地址 2—位地址

不少厂家的 PLC 地址及内存的容量是以字节计，单独用它的位地址时，是用字节地址代替这里的字地址。如西门子 PLC 输入点编号要先加 I（用英文）或 A（用德文），输出点要先加 Q（用英文）或 O（用德文）。施耐德及国产和利时 PLC 输入点要先加% I，输出点要先加% O。然后是字节地址，继而才是“位”地址。

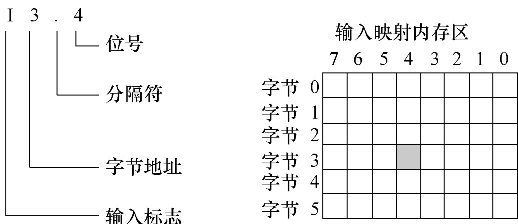


图 2-12 字节地址与位地址

如图 2-12 所示的地址 I3.4，这里 I 的含义是输入点，3 为字节地址，4 为字节 3 中的第 4 位。

但是，有的厂家的 PLC 没有这个字或通道概念，它以位（bit）编地址（其数据存储器以字编址，但不能按位使用）。如三菱 FX 机的 M 区，从 M0 到可能的最大地址，都是以位为单位编址。并按八（对小型机）或十六（对中、大型机）进制进位。但没有字或字节地址。当多位使用时，则应在其前，加上使用多少位的指定。如 K1X0，指的是 X0 ~ X4，4 个位。再如 K2X0，指的是 X0 ~ X7，8 个位。其余类推。这样，如按字节或字使用，虽较麻烦，但这些点的使用，可不受字节、字的局限，也有其灵活性。

总之，PLC 数据区地址编号及表达都有一定的规律，与生产厂家、PLC 的型号及使用设定有关。

2. 数的表示

用字、字节还可以表示数。字表示为十六进制数时，最大的数值为 FFFF，折合成十进数为 65535。十六进制数还可表达带符号位的数。最高位为符号位，1 表示负数时，0 表示正数。其它位用以表示数，负数用补码表示。正数也用补码。只是正数的补码与原码相同。表示带符号位的数的范围为-32768 ~32767。见表 2-3。

表 2-3 还示出数用 BCD 码表达的情况。这时，每数位最大只能到 9，超过 9，则变为 0，并向上进位。一个字最大的可表达的值为 9999，折合成十进数也是 9999。用 BCD 码表达数

表 2-3　不同码制数的表达

数据类型	数　据　格　式	十进制	表达数的范围
十六进制码	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 二进制 2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^{9} 2^{8} 2^{7} 2^{6} 2^{5} 2^{4} 2^{3} 2^{2} 2^{1} 2^{0} 十进制 32768 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1 十六进制 2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0	0 ~ 65535	0000 ~ FFFF
带符号位的十六进制码	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 二进制 2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^{9} 2^{8} 2^{7} 2^{6} 2^{5} 2^{4} 2^{3} 2^{2} 2^{1} 2^{0} 十进制 32768 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1 十六进制 2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0 符号位　0 为正　1 为负	−32768 ~ +32767	8000 ~ 7FFF
BCD 码	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 二进制 2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0 十进制 0～9 0～9 0～9 0～9	0 ~ 9999	0000 ~ 9999

容易理解，符合常人的习惯。早期 PLC 功能差，数的处理、运算，工作量不大，BCD 码用得很多。但用 BCD 码表达的值范围小，存储器的资源不能得到充分利用。而且，它不能表达所有位的值（如 9 以上的值），所以，其使用是受限制的。

提示：BCD 码也称二十进制码。数位是二进制，由 4 个二进制数组成。而数位之间又为十进制，每数位只能在 0（0000）~9（1001）之间取值。十六进制计码，数位内与数位之间都是二进制。每数位可在 0（0000）~F（1111）之间取值。显然，如把这两码制不同的数作比较，就有可能出现永远不相等的情况。

数据处理时，要表达比 65535 更大的数，或用 BCD 码数表达的数大过 9999，则要用两个字，即双字长。它有 4 个字节，8 个数位，32 个二进制位。

用双字表达一个数时，以低字（或最低字节，对地址以字节计的 PLC）的地址代表它的地址。一般高地址的字、字节存放高位数。但西门子的 PLC 不同，最低的字节存的为最高位的数。使用时要注意这个区别。

双字还可表达为浮点数（带符号位）。还可用 4 个字表达双精度的浮点数（也带符号位）。其表达格式与普通计算机的表达格式完全相同，见表 2-4。表达为浮点数时，一般可不必了解它的各二进位的细节。浮点数与十六进制数、BCD 码数，都有相应的相互转换指令。当然，应确保在转换后，结果数不能溢出，能在表达的值范围内。

表 2-4　浮点数表达格式

数据类型	数　据　格　式
单精度浮点数	31 30 29 … 23 22 21 20 19 18 17 … 3 2 1 0 符号位　指数　尾数 值 = $(-1)^{符号位} \times 1.尾数 \times 2^{指数}$

（续）

数据类型	数据格式
双精度浮点数	63 62 61 … 52 51 50 49 48 47 46 … 3 2 1 0 符号位　指数　尾数 值 $=(-1)^{\text{符号位}}\times 1.\text{尾数}\times 2^{\text{指数}}$

3. PLC 常数

PLC 常数有 16 位的，也有 32 位的，见表 2-5。要注意，输入常数一定要加上相应的前缀，如“#”（十六进制数）、“&”（十进制数）等。如常数为实数，即浮点数，则要加小数点。即使小数点后的值为 0，也要加“. 0”。

表 2-5　不同格式的常数表达

方式	使用的操作数	数据格式	代码	范围
常数(16 位数据)	所有二进制数据和一定范围之内的二进制数据	不带符号的二进制	#	#0000 ~ #FFFF
		带符号的十进制	±	−32768 ~ +32767
		不带符号的十进制	&	&0 ~ &66535
	所用 BCD 数据和一定范围之内的 BCD 数据	BCD	#	#0000 ~ #9999
常数(32 位数据)	所有二进制数据和一定范围之内的二进制数据	不带符号的二进制	#	#0000 0000 ~ #FFFF FFFF
		带符号的十进制	±	−2147483648 ~ +2147483647
		不带符号的十进制	&	&0 ~ &4294967295
	所有 BCD 数据和一定范围之内的 BCD 数据	BCD	#	#00000000 ~ #99999999
常数(32 位浮点数据)	所有 32 位浮点数据	带符号及小数点的数	±x、y	-3.402823×10^{38} ~ 1.175494×10^{-38},0, $+1.175494\times10^{-38}$ ~ $+3.402823\times10^{38}$
常数(64 位浮点数据)	所有 64 位浮点数据	带符号及小数点的数	±x、y	$-1.79769313486232\times10^{308}$ ~ $2.22507385850720\times10^{-308}$, 0, $2.22507385850720\times10^{-308}$ ~ $1.79769313486232\times10^{308}$

提示：弄清数据格式是很重要的。如用的 BCD 码运算指令，若操作数为十六进制格式，则指令将不执行。

除了存数，PLC 内部器件还可存字符，一般一个字节存一个字符，按十六进制 ASCII 码存。

4. PLC 软器件概况

PLC 的软器件有两大类：入出器件及内部器件。

（1）入出器件也可称映射器件，是内存区中与输入、输出模块相映射的区域。与输入点、输入通道对应的称输入继电器、输入通道，或称输入映射区。与输出点、输出通道对应的称输出继电器、输出通道，或称输出映射区。

如用有模拟量模块或其它智能模块，则还包括与其有关的数据映射区。如 PLC 联网，则还含有与联网有关的数据链接区。

（2）内部器件用于存储中间结果的内存区。内部器件与硬件没有映射关系，可自由使用。OMRON 公司 PLC 的内部器件有：内部辅助继电器、特殊继电器、保持继电器、计数器、定时器、链接继电器（如不用于联网链接）、数据存储器、暂存器等等。

软器件的类型及数量，反映了 PLC 的控制能力，是 PLC 性能的重要方面。越是性能好的 PLC，越是新型的 PLC，其类型也越多，数量也越大。

PLC 技术的发展，其软器件的类型及数量，特别是内部器件现已大为增加。所以，OMRON 新型的 PLC 已不用继电器（Relay）称谓它了。而改称它为区（Area），突出了 PLC 信息处理的功能。

以 CJ1 机为例，则分为：核心 I/O（CIO）区（即这里将要介绍的入出器件）、工作区、保持区、辅助区、DM 区、EM 区、定时器区、计数器区、任务标志区、数据寄存器、索引寄存器、条件标志区、时钟脉冲区等。表 2-6 表示了 CJ1 机的数据区结构。

表 2-6　CJ1 机内存区分配

内存区		大小	范围
CIO 区	I/O 区	1280 bit(80 word)	CIO 0000 CIO 0079
	Data Link 区	3200 bit(200 word)	CIO 1000 ~ CIO 1199
	CPU Bus Unit 区	6400 bit(400 word)	CIO 1500 ~ CIO 1899
	Special I/O Unit 区	15360 bit(960 word)	CIO 2000 ~ CIO 2959
	DeviceNet 区	9600 bit(600 word)	CIO 3200 ~ CIO 3799
	Internal I/O 区	37504 bit(2344 word) 4800 bit (300 word)	CIO 1200 ~ CIO 1499 CIO 3800 ~ CIO6143
工作区		8192 bit(512 word)	W000 ~ W511
保持区		8192 bit(512 word)	H000 ~ H511

内存区	大小	范围
TR 区	16bit	TR0 ~ TR15
DM 区	32768 word	D00000 ~ D32767
EM 区	32768 word 每段 (0 ~ 2. 3max.)	E0_00000 ~ E2_32767
定时完成标志	4096 bit	T0000 ~ T4095
计数完成标志	4096 bit	C0000 ~ C4095
Timer PV 定时器现值	4096 word	T0000 ~ T4095
Counter PV 计数器现值	4096 word	C0000 ~ C4095
任务标志	32 bit	TK00 ~ TK31
变址寄存器	16 register	IR0 ~ IR15
数据寄存器	16 register	DR0 ~ DR15
辅助区	15360 bit(960 word)	A000 ~ A959

2.2.2 入出器件

不同的 PLC，其入出器件是很不相同的，以下主要以 CJ1 机为例介绍这类器件。CJ1 机的入出器件基本上是指它的核心内存区（CIO）。它含有 I/O 内存区，特殊 I/O 内存区，CPU 总线内存区，数据链接内存区及内部 I/O 内存区。其分布如图 2-13 所示。

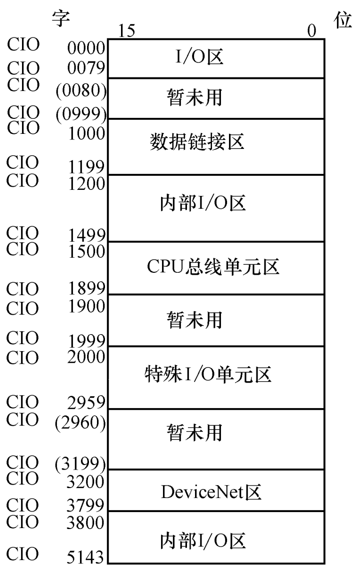


图 2-13 入出器件地址分布

1. 开关量用的输入、输出继电器

开关量用的输入、输出继电器（也称 I/O 继电器，对 CJ1 机即为 I/O 数据区，是指可与实际输入、输出点对应的那部分内存区。它决定了 PLC 可能配置的最多 I/O（开关量）点数。

输入继电器与输入点对应。当 PLC 运行到输入刷新阶段时，输入暂存器的状态即映像到输入继电器中。输入继电器多以位为单位被读出，也可以字（通道）为单位传递、译码。输入继电器为只读存储器，不能用程序改变它的内容，而只能被输入点所映像。所以无输入点与其对应的，如其地址是固定编排时，一般不能作为它用。

输出继电器与输出点对应。当 PLC 运行到输出刷新阶段时，输出继电器的状态被映像到输出内部电路的锁存器。锁存器把状态保持，直到下一个刷新的到来。锁存器再经输出电路传递，即成为输出点上的输出。输出继电器是可写的，以便产生所要求的输出；也还是可读的，以用于反馈控制。对用户程序，它是可读可写的存储单元。所以，若无输出点与其对应的也可另作它用。

西门子 PLC 称之为过程映射寄存器（Process-Image Register），指的也是与输入、输出点有对应关系的内存区。

使用 PLC，一定要了解输入、输出继电器的情况。要清楚，它有多少输入、输出继电器，怎么编号，与输入、输出点的物理位置怎么对应的（在模块上有相应标号）等，否则无法编程，也无法接线。

前已指出，输入、输出继电器编号一般先编通道号（高三位或四位），再编位号（低两位）。所以，它的地址一般为五位或六位位数。

OMRON 公司的 CJ1 机（以下简称 CJ1 机）I/O 继电器编号为六位数。头四位为通道号，后两位为点号。点在模块中的编号是固定的，用两位数，与模块上标明的点号是对应的。一个模块如只有 16 个点，或更少，则一个模块占一个通道号。点数超过 16、少于 32 的占两个号，超过 32、少于 64 的占四个号。

通道编号是按模块所在机架的位置及模块在机架上的位置决定。先编 CPU 机架上最靠近 CPU 的模块（C200H 系列机，CPU 在机架的右方，所以，先编最远离 CPU 的模块），接着编机架上其它模块，从左到右，依次升幂编。CPU 机架上的模块编好后，再编最靠近 CPU 机架的扩展机架上的各模块，也是从左到右，依次升幂编。直到编完最远离 CPU 机架的扩展机架上的最右边的一个模块，如图 2-14 所示。

这种编号，次一机架的最左一个模块号，与前一个机架的最右一个模块号，总是顺序衔接的。这是默认的编号方法。

为了方便，也可对每一机架最左边的模块自行指定一个模块号，然后其它的依次升幂编。但不管怎么指定，应确保每一模块号是惟一的。并且，所有的模块号都应落在 000 ~079 之间。

图 2-15 所示为 CJ1 单机架配置时的输入、输出继电器编号。它的机架编号用默认的。该系统用了 3 个输入模块，其中两个为 16 点的，一个为 32 点的。两个输出模块，其中一个为 32 点，一个为 64 点。该图分别对各模块都标明了相应的地址。表 2-7 对图 2-15 的 I/O 模块编址也作了注解。

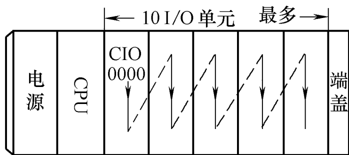


图 2-14　CPU 机架地址编定

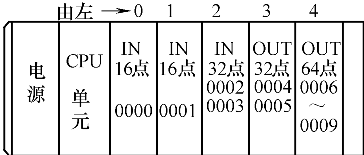


图 2-15　CPU 机架地址编定例

表 2-7　I/O 模块编址

位置	模　　块	占通道数	分配地址
0	CJ1W-ID211 16 点　DC 输入单元	1	CIO 0000
1	CJ1W-ID211 16 点　DC 输入单元	1	CIO 0001
2	CJ1W-ID231 32 点　DC 输入单元	2	CIO 0002 和 CIO 0003
3	CJ1W-OD231 32 点　晶闸管输出单元	2	CIO 0004 和 CIO 0005
4	CJ1W-OD261 64 点　晶闸管输出单元	4	CIO 0006 ~ CIO 0009

图 2-16 所示为 CJ1 机多机架配置时的输入输出继电器编号例子。该系统用了 3 个机架。共用了 11 个模块。其清单见表 2-8。

表 2-8　多机架地址编定

机架	槽号	单　　元	要求字	字地址
CPU 机架	0	CJ1W-ID211 16 点　DC 输入单元	1	CIO 0000
	1	CJ1W-ID231 32 点　DC 输入单元	2	CIO 0001 和 CIO 0002
	2	CJ1W-ID261 64 点　DC 输入单元	4	CIO 0003 ~ CIO 0006
	3	CJ1W-OD211 16 点　晶闸管输出单元	1	CIO 0007
	4	CJ1W-OD231 32 点　晶闸管输出单元	2	CIO 0008 和 CIO 0009
扩展机架	0	CJ1W-ID211 16 点　DC 输入单元	1	CIO 0010
	1	CJ1W-ID231 32 点　DC 输入单元	2	CIO 0011 和 CIO 0012
	2	CJ1W-OC201 8 点　继电器输出单元	1	CIO 0013
扩展机架	0	CJ1W-ID211 16 点　DC 输入单元	1	CIO 0014
	1	CJ1W-ID231 32 点　DC 输入单元	1	CIO 0015 和 CIO 0016
	2	CJ1W-OC211 16 点　继电器输出单元	1	CIO 0017

如图 2-16 所示，其 CPU 机架号为 00（或说，它的第一个模块编号为 0000）。在 CPU 机架上，安装 5 个模块，3 个输入，2 个输出。

扩展机架两个。靠近 CPU 的机架的有两个输入模块、一个输出模块。后一个扩展机架也有两个输入模块、一个输出模块。

这里，次一机架的最左一个模块号，都没有另行编号指定，故它们的编号全部是连续的。各模块具体编号见该图模块上的标注。

CJ1 机多机架配置时，要用到接口模块及相应连接电缆，这里未标出。

开关量用的输入输出点的地址编号，除了这里介绍的，还有 3 种办法：

（1）点地址固定，用于箱体式 PLC。其主箱体、扩展箱体连接好后，其各点的地址也就确定了。

（2）模块地址固定，模块安装好后，其地址依其所在机架及其上的位置，也默认地确定了。点在模块上的地址自然也是固定的。

（3）模块地址自行设定，这已是发展趋势。

具体怎么编址，要按有关说明书操作。

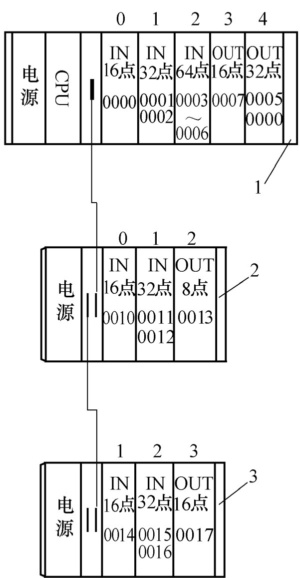


图 2-16　多机架地址编定

1—CPU 机架　2、3—扩展机架

2. 模拟量输入输出通道

除了开关量 I/O 模块，还有特殊 I/O 模块。对 OMRON 公司的 PLC，它的编号与开关量 I/O 模块无关，按自身规定排列。

它的特殊 I/O 模块都有自己的设定机号。这由其上的设定开关设定。CJ1 机可在 0 ~ 94 之间选定。设定后，系统将为每一模块提供 10 个 I/O 通道及 100 个字的数据存储区（DM 区，见后）为其专用。功能特强的特殊模块，如位控模块，系统将为每一模块提供 20 个 I/O通道及 200 个字的数据存储区（DM 区，见后）为其专用。

要注意的是，所设的机号不能重复。如有的占 20 个通道、200 个字的 DM 地址的，次一个机号要隔开。一定保证地址不重复。

这意味着，CJ1 机最多可控制 95 个特殊 I/O 模块。

CJ1 机占用通道与 DM 的地址与设定的机号关系为

如机号为 N，则占用的通道号为：

$$2000 + 10 * N \sim 2000 + 10 * N + 9$$

占用的 DM 地址为

$$DM20000 + 100 * N \sim DM20000 + 100 * N + 99$$

至于这些通道与 DM 区的什么含义及怎么使用，则所选用的模块确定。要参考有关说明书。以上公式也只适合 CJ1 机。别的机型又有别的公式，这里略。

图 2-17 所示为在一个 CJ1 机 CPU 机架上安装 5 个模块的情况。表 2-9 表示了它的各个槽位及相应模块地址分配。

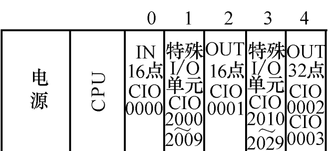


图 2-17　单机架地址编定例

表 2-9　单机架地址编定例

槽	单　　元	要用的字	字地址分配	单元号	类别
0	CJ1W-ID211 16 点　DC 输入单元	1	CIO 0000	—	基本 I/O 单元
1	CJ1W-AD081　模拟输入单元	10	CIO 2000 ~ CIO 2009	0	特殊 I/O 单元
2	CJ1W-OD211-16 点　晶闸管输出单元	1	CIO 0001	—	基本 I/O 单元
3	CJ1W-TC001　温度控制单元	20	CIO 2010 ~ CIO 2029	1	特殊 I/O 单元
4	CJ1W-OD231 32 点　晶闸管输出单元	2	CIO 0002 和 CIO 0003	—	基本 I/O 单元

西门子公司 S7-300 PLC 模拟量模块与开关量模块都称为信号模块，不算什么特殊 I/O 单元，无单元号。这两种模块都依其所在机架及槽位统一编址。只是模拟量加的前缀为 IW（模入）或 QW（模出），每一路要占一个字、两个字节。

表 2-10 表示西门子 S7-300 机，如配置 4 个机架时的编址情况。从图 2-17 知，它是定机架、定槽位编址的。如有一 8 路模入模块安装在机架 1、槽位 6，其各路地址分别为 IW416、IW418……直到 IW430。如这槽位安装的是 4 路模出模块，则其各路地址分别为 QW416、QW418、QW420 及 QW422。如这槽位安装的是 16 点开关量输入模块，则其地址为 IB40、IB41 两个字节。

表 2-10　S7-300 机地址分配

机架	模块开始地址	槽号										
		1	2	3	4	5	6	7	8	9	10	11
0	开关量模块 模拟量模块	电源	CPU	接口 模块	0 256	4 272	8 288	12 304	16 320	20 336	24 352	28 368
1	开关量模块 模拟量模块	— —		接口 模块	32 384	36 400	40 416	44 432	48 448	52 464	56 480	60 496
2	开关量模块 模拟量模块	— —		接口 模块	64 512	68 528	72 544	76 560	80 576	84 592	88 608	92 624
3	开关量模块 模拟量模块	— —		接口 模块	96 640	100 656	104 672	108 688	112 704	116 720	120 736	124 752

3. CPU 总线模块及其工作区

CJ1 机的 CPU 总线内存区在 CIO 区的字地址，是从 1500 ~ 1899，共 400 个字，供 CPU 总线模块使用。每个 CPU 总线模块要用 25 个字。这意味着，CJ1 机最多只能配置 16 个 CPU 总线模块。

每一 CPU 总线模块要设一个惟一的机号。由模块上的设定开关设定。机号分别设为 0 ~ 15。机号设定后，其工作使用的字地址也就确定了。表 2-11 示出机号与地址的对应关系。

表 2-11　单元号与 CIO 地址

单元号	分配地址	单元号	分配地址
0	CIO 1500 ~ CIO 1524	⋮	⋮
1	CIO 1525 ~ CIO 1549	15	CIO 1875 ~ CIO 1899
2	CIO 1550 ~ CIO 1574		

CPU 总线模块工作时，还要用到 DM 区（见后）。设定机号后，系统为每一模块预留了 100、200 或 400 个字。其地址也与设定的机号有关。见表 2-12。

表 2-12　单元号与 DM 区地址

单元号	分配地址	单元号	分配地址
0	DM30000 ~ DM30099	⋮	⋮
1	DM30100 ~ DM30199	15	DM31500 ~ DM31599
2	DM30200 ~ DM30299	—	—

图 2-18 所示为 CJ1 机配置有基本 I/O 单元、特殊 I/O单元及 CPU 总线单元的编址情况。

从图 2-18 知，它用了两个基本 I/O 单元，一个特殊 I/O 单元及两个 CPU 总线单元。具体单元号设定及字地址分配见表 2-13。

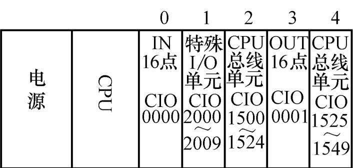


图 2-18　多种单元配置编址

表 2-13　多种单元配置编址

槽	单　　元	要用的字	字地址分配	单元号	类别
0	CJ1W-ID211 16 点　DC 输入单元	1	CIO 0000	—	基本 I/O 单元
1	CJ1W-AD081　模拟输入单元	10	CIO 2000 ~ CIO 2009	0	特殊 I/O 单元
2	CJ1W-SCU41 系列通信单元	25	CIO 1500 ~ CIO 1524	0	CPU 总线单元
3	CJ1W-OD211 16 点　晶体管输出单元	1	CIO 0001	—	基本 I/O 单元
4	CJ1W-CLK21 控制器链接单元	25	CIO 1525 ~ CIO 1549	1	CPU 总线单元

4. 其它工作区

除了上述，还有数据链接区（Data Link）、Devicenet 网的数据区。前者，用于网络数据链接，后者用于 Devicenet 网配置。有的 OMRON PLC 还有远程配置，其远程单元的编址，还要用到远程地址区。

数据链接区，过去称链接继电器（Link Relay），用于联网时的数据链接。

OMRON 小型机链接继电器有 64 个通道，地址为 LR00 ~ LR63。可按字使用，也可按位使用。使用时要加前缀“LR”。

CJ1 机数据链接区有 1200 个字，可进行交换的数据量要大的多。其地址从 1000 ~ 1199，使用时不加前缀。

最后，还要在这里介绍一下 OMRON 中、大型 PLC 的 I/O 表登记。

I/O 表是对 PLC 各个模块安装、设定与编址的确认。I/O 表登记就是把这些确认的数据下载给 PLC，并存入 PLC 的设定区。I/O 表可以不登记，即不下载。这样，每当 PLC 上电，系统将与小型机一样，按各模块实际的安装、设定与连接，确定各模块地址。

如果作了 I/O 表登记，则每当 PLC 上电时，系统将检查实际的模块安装、设定与连接，看其是否与已登记的 I/O 表一致。如不一致，将提示 I/O 总线错（I/O BUS ERR）。PLC 将

不能运行程序。显然，I/O 表登记的目的是对现场的安全保护。因为，如果模块安装错了，系统按安装错了的地址运行程序，那是很危险的。

I/O 表登记可用编程软件进行。用软件确定各机架上安装的所有 I/O、特殊 I/O 及 CPU 总线单元的特性。确定后，下载给 PLC；也可用手持编程器进行，用操作命令做也很简便。具体见编程软件及编程器使用的介绍。

2.2.3　内部器件

不同的 PLC，其内部器件也是很不相同的，以下主要以 CJ1 机为例，介绍这类器件。

1. 内部辅助继电器

它与输入点、输出点无对应的映射关系，但可通过相应指令与输入、输出继电器建立一定的逻辑联系。与继电器电路中的中间继电器一样，运用得合理，可帮助实现输入与输出间复杂的变换，从而可使 PLC 更好地进行种种控制。

内部辅助继电器的数量一般要比输入、输出通道多得多。除了用作中间继电器，还用于数据处理。

内部辅助继电器也分配有通道号。每个通道也是 16 个继电器。可以通道（字）为单位使用，也可以继电器（位）为单位使用。OMRON 公司 CJ1 机的内部辅助继电器包含有内部 I/O 区（地址不带前缀）及工作区（地址带前缀 W），共有几千个字，几万个位。而且，无输入、输出点对应的输入、输出继电器也可用作内部辅助继电器，其编号仍用原继电器的编号。

不同公司或不同系列的 PLC 的内部辅助继电器，其数量及编号也不完全相同。如西门子公司、施耐德公司、三菱公司 PLC 的内部继电器，就加有前缀“M”。而且，是以字节为单位编号。

使用 PLC 时，对选用的 PLC 的内部辅助继电器的情况也要弄清楚，否则也无法编程。它的数量非常多，编程时对它的使用并不感到困难。

2. 保持继电器

CJ1 机称为保持区。它与输入点、输出点也无对应的映射关系。但它可掉电保持，即 PLC 掉电或工作模式改变时，其内容保持不变。使用保持继电器，在一定程度上可使 PLC 少受掉电影响，保证程序运行的连续性。

OMRON CJ1 机的保持继电器加有前缀，为 HR。有 512 个通道，HR000 ~ HR511，每个通道有 16 个继电器，共 4096 个。

CJ1 机还可通过设定（IO 存储保持位，A50012（见后），置 1 及 PLC 起动模式设为保护 IO 保持位），做到内部辅助继电器在掉电后，数据也能保持。

多数公司无此继电器，但它们的数据区多可设置成掉电保持的。显然，既可作这样的设定，就不必有这类单独的保持继电器了。

3. 定时器

它与继电电路的时间继电器类似，用于延（定）时控制。它有线圈，有触点（标志位），还有寄存器（存放定时器现值）。定时的设定值可为常数，也可为地址（通道号、存储器号），再用地址的内容作为设定值。以 CJ1 机为例，当定时器的线圈 OFF 时，没有输出，其寄存器的当前值为设定值，其常开触点为 OFF，常闭触点为 ON。当定时器的线圈 ON

时，它的寄存器的当前值从设定值开始定时往下减（每单位设定值时间减 1）。减到零时，即产生输出，其常开触点从 OFF 转为 ON，常闭触点从 ON 转为 OFF。任何时候，一旦其线圈 OFF，其输出立即停止，其常开触点从 ON 转为 OFF，常闭触点从 OFF 转为 ON，寄存器的当前值又变为设定值。

CJ1 机的定时器有 4096 个，编号从 TIM0000 ~ TIM4095。其中的 0000 ~ 2027，在执行跳转指令期间，它的现值仍可更新，继续工作。

定时器设定值可用 4 位十进制（BCD 码），设定范围为 0000 ~ 9999。也可用十六进制码，设定范围为 0 ~ 65535。普通定时器单位设定值为 0.1s，故其最大延时可达 999.9s，或 6553.5s。如处理成高速定时（使用相应的高速计数指令），其单位设定值可能为 0.01s、0.001s，故其最大延时为 99.99s、9.999s，或 655.35s、或 65.535s。如处理成低速定时（使用相应的低速处理计数指令），其单位设定值可能为 1s 或 1min，故其最大延时为 9999s、9999min，或 65535s、65535min。

所有 OMRON 的 PLC 的定时器都是 ON 延时的，而 OFF 是即时的。若要求 OFF 延时，要用编程或使用系统功能块（见后）解决。

应该指出，PLC 的定时器的定时控制是运行程序实现的。由于输入响应延时及扫描工作方式的影响，定时控制不是很准确，可能与设定值差一个扫描周期。扫描时间若大过单位设定值，只有若干个定时器（编号低的，可中断工作的）才能准确工作。

还要指出，PLC 的定时器多为掉电不保持的，掉电后停止计时，已计值不保留，复电时，再从头计时。但有的 PLC，如 CJ1 机可通过设定（IO 存储保持位，A50012，置 1 及 PLC 起动模式设为保护 IO 保持位），做到掉电保持或 PLC 工作方式定时器数据保持。

提示：定时器怎么用，与相应的定时指令的使用有关，这在介绍定时指令时，还要作进一步说明。

4. 计数器

它与继电电路用的计数器类似，用于记录脉冲输入信号从 OFF 到 ON 的次数。它有线圈，有触点（标志位），还有寄存器（存放计数器现值）。有两种计数：一为单向计数；另一为可逆（双向）计数。

（1）单向计数。使用单向计数指令。开始时，计数器现值为设定值。送入计数信号，计数器现值减 1。减到零，则产生输出。其常开触点 ON，常闭节点 OFF。

产生输出后，再送入计数信号，计数器现值及触点状态不变。任何时候，送入复位信号，计数器现值都恢复成设定值，并停止计数，输出停止。

有的厂家 PLC 单向计数为增计数，复位后或开始时，计数器内容为 0。来一次计数信号，计数器增 1，达到或超过设定值时，产生输出，而且，仍可继续计数。

（2）可逆计数。要使用可逆计数指令，开始时，计数器现值为 0。送入增计数信号，计数器现值加 1；送入减计数信号，则减 1。增计数到设定值，再送入一个增计数信号，或减计数到零，再送入一个减信号，会产生计数进位或借位，并产生输出。其常开触点 ON，常闭触点 OFF。复位信号 ON，计数器现值变成零，并停止计数。

CJ1 机计数器标号为 CNT。其编号为 0000 ~ 4095，共 4096 个。计的数可用 4 位十进制（BCD 码），也可用十六进制码，由用什么计数指令决定。计数器都是掉电保持的，掉电后计数值保留，复电后在原计数值基础上继续计数。

可掉电保持的计数器与产生定时（如 001s 的）脉冲信号的特殊继电器配合，也可构成积算式的定时器，可用于累计计时。

提示：计数器与计数指令的使用有关，在介绍计数指令时，还要作进一步说明。

5. 数据存储区（DM）

以前称为数据存储器。PLC 进行控制，总要作一些数据处理，使用特殊单元，也要作数据计算。所以，各型 PLC 都有专门存储数据的单元。OMRON 公司 PLC 的数据存储器的标号为 DM。可单字使用，也可双字、多字使用。CJ1 机还可按位作逻辑处理。

CJ1 机 DM 区有达 32K 字，而且可间接寻址。但其中被特殊或 CPU 单元使用（如安装有这类单元），其它不能用。如图 2-19 所示。

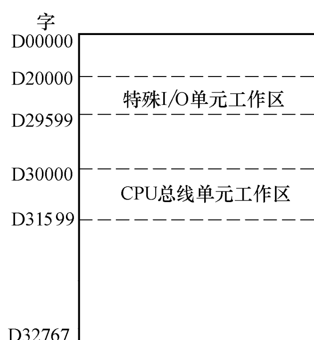


图 2-19　CJ1 机 DM 区

数据存储区为掉电保持的。

OMRON 小型机的 DM 区小，才几 K 字。其中，还有些字用作系统设定。如 CQM1 等机的 DM6600 ~ DM6655，共 56 个字，就是做这个用途的。其中：

DM6600 ~ 6614：用作起始处理设定；可设定 PLC 起始模式为编程、运行或监控模式及内部继电器上电时是否清零。

DM6615 ~ 6619：用作脉冲输出及扫描周期的设定。

DM6620 ~ 6639：用作中断处理设定。

DM6640 ~ 6644：用作高速计数设定。

DM6645 ~ 6649：用作对 RS232 口的设定。

DM6650 ~ 6654：用作对外设口的设定。

DM6655：用作出错记录设定。

CJ1 机 DM 区中无此设定字。它的系统设定，另有设定区。

西门子 PLC 的数据区为数据块，使用前要先设定，前缀为 D，如 DB、DW、DD，分别代表字节、字及双字。数据块也可按位使用。它的小型机的数据区前缀为加 V。如 VB、VW、VD，分别代表字节、字及双字。它的所有数据区均以字节计地址。如用 VW0 地址，要用到字节 0 与字节 1。其后再有地址必须为 VB2，或 VW2，或 VD2，而不能用 VB1。

6. 扩展数据存储区（EM）

它是 DM 区的补充或扩展。是多用途的数据存储区，只能以字为单位使用。它是掉电保持的，当 PLC 掉电或从工作模式改变，其内容保持。

EM 区依 32767 字为单位划分成若干段（Bank，视 PLC 的型别而定）。目前，CJ1 机最多的为 13 段。在这么多段中，只有一个为当前段。只有在当前段的数据可用通常的方法使用，但当前段可用指令改变。

EM 区的编址有两种方法：一为先指出段地址，后指出在段中的地址；另一为只指出在段中的地址，这仅仅是对当前段而言。如 E2_00100，指的是段 2，第 00100 字。如 E00100，则指的是当前段，第 00100 字。

CJ1 机的部分 EM 区还可设定为文件转换区（OMRON 以前的 PLC 不能这么做）。它以文件的形式存储数据。而一旦这些区设为文件转换区，它就不能用普通的数据处理指令访问。

7. 特殊继电器

它也是一种内部辅助继电器，只是各有其特殊用途。

OMRON 的小型机特殊继电器有 12.5 通道，200 个继电器。其编号从 244 开始，直到 255 的前半个通道。常用的如：

253.13：常 ON 触点；

253.14：常 OFF 触点；

253.15：进入运行或监控状态后 ON 一个扫描周期的触点；

254.00：1 分脉冲，0.5 分 ON，0.5 分 OFF；

255.00：0.1 秒时钟脉冲，0.05 秒 ON，0.05 秒 OFF；

255.01：0.2 秒脉冲，0.1 秒 ON，0.1 秒 OFF；

255.02：1 秒脉冲，0.5 秒 ON，0.5 秒 OFF；

255.03：指令执行出错标志，指令出错 ON；

255.04：进位标志，执行指令有进位时 ON；

255.05：大于标志，作比较，当第一操作数大于第二操作数时 ON；

255.06：相等标志，比较相等，或结果为零时 ON；

255.07：小于标志，作比较，当第一操作数小于第二操作数时 ON；

由于通道太多，不好一一在此列出，可参阅有关说明。

OMRON 的 C200Hα 型机，其特殊继电器又增加了不少，其起始通道为 236，终了通道为 299，共 64 个通道，共 1024 特殊继电器。这么多的特殊继电器，与 C200Hα 型机有强大的控制能力是相适应的。而早期的 PLC，如 OMRON 的 P 型机，特殊继电器仅 16 个，比这要少得多。

CJ1 机没有特殊继电器的称谓，而用标志（Condition Flag，CF）及辅助区（AR）中的一些字与位起到这里讲的特殊继电器的作用。如这里的 25506（相等标志），用标志 CF006 代替。

CJ1 机标志是一些指令执行的结果，或有固定含义。所以，它是只读的，不能用程序或编程软件改写。CJ1 机主要的标志有（前缀为“CF”的为标志，前缀为“A”的为辅助继电器）：

CF1.13：常 ON 触点；

CF1.14：常 OFF 触点；

A500.15：进入运行或监控状态后 ON 一个扫描周期的触点；

CF1.00：0.1 秒时钟脉冲，0.05 秒 ON，0.05 秒 OFF；

CF1.01：0.2 秒脉冲，0.1 秒 ON，0.1 秒 OFF；

CF1.02：1 秒脉冲，0.5 秒 ON，0.5 秒 OFF；

CF1.04：1 分脉冲，0.5 分 ON，0.5 分 OFF；

CF0.03：指令执行出错标志，指令出错 ON；

CF0.04：进位标志，执行指令有进位时 ON；

CF0.05：大于标志，作比较，当第一操作数大于第二操作数时 ON；

CF0.06：相等标志，比较相等，或结果为零时 ON；

CF0.07：小于标志，作比较，当第一操作数小于第二操作数时 ON；

也由于通道太多，不好一一在此列出，可参阅有关说明。

8. 辅助继电器（AR）

OMRON 的小型机，只有 28 个辅助继电器通道，共 448 个继电器。编号为 AR00 ~ AR27。多数 AR 继电器有特殊用途。只有其中的部分通道，如 AR00 ~ AR07 未被指定，可作为内部辅助继电器使用。

CJ1 机辅助区有 960 个字，8192 个位，如图 2-20 所示。使用它要加前缀“A”。A000 ~ A477 为 PLC 工作标志字或位，为只读的。A448 ~ A959 为 PLC 工作控制字或位，是可读写的。

它虽与输入点、输出点无对应物理关系，但它与 PLC 的工作息息相关。是 PLC 非常重要的内存工作区。

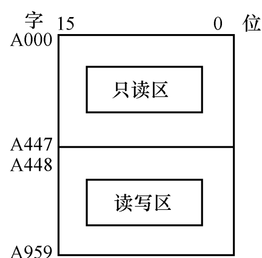


图 2-20　CJ1 机辅助区

9. 暂存器（TR）

它仅能用作 LD、OUT 指令的操作数。类似堆栈，仅作暂存，用以处理梯形图中有分叉的输出。在程序中可多次使用，只要不引起误会即可。CJ1 机有 16 个位的暂存器，而 OMRON 老的 PLC 自有 8 个位。

它的增加，意味着可处理更为复杂的梯形图程序。

用梯形图编程时，用户已看不到到这类暂存器的使用。若把所编的程序自动转换成助记符程序时，将看到它的使用。或用助记符编程时就要使用它。

10. 索引寄存器（Index Register）

它可用作指针，用它的值指向 PLC I/O 内存区的地址，以对数据区作间接访问。CJ1 机共有 16 个索引寄存器，字长为 32 位，地址从 IR00 ~ IR15，所加的前缀为“IR”。

提示：索引寄存器为双字长，存储的地址为绝对地址（Absolute Memory Addresses in I/O memory）。 可存字（word）地址，也可存位（bit）地址。存位（bit）地址时，高 7 个数位（digit）存放字（通道）地址，其它数位存放位地址。对其赋值，不能简单地用 MOV 之类指令，要用地址赋值指令，即 MOVR（560）指令。读定时器现值（PV）的地址要用 MOVRW 指令。

过去 OMRON 的中、小型机的 DM 区也可间接访问，而有了这个索引寄存器，间接访问可以扩大到整个 I/O 内存区。

11. 数据寄存器（Data Register）

CJ1 机还有 16 个数据寄存器，所加的前缀为“DR”，地址从 DR00 ~ DR15，字长为 16 位。它常与索引寄存器配合使用。它可使用数据处理指令赋值或重新赋值，但赋值后不能用编程软件改变。

各厂家、各型号 PLC 的内部器件不完全相同，有的不完全有上述 11 个方面，有的还有别的什么名称。如三菱 PLC 的 FX2 机还有状态继电器，编号为 S000 ~ S999 共 1000 点，用作步进控制。

OMRON 的 CV 机可用流程图语言编程（见后），所以，还有：

转移区（Transation Area），TN0000 ~ TN1023 共 1024 个字，用作流程图编程时转移标志。

步进区（Step Area），ST0000 ~ ST1023，也是 1024 个字，用作流程图编程的步标志。

等等。

了解了 PLC 的软器件，即 PLC 的数据存储区，也就了解了 PLC 指令的操作数。至于这些操作数怎么被指令操作，将在介绍 PLC 的指令时，作具体介绍。

2.3 PLC 指令系统

关键词：指令系统、操作数、即时数、直接地址，间接地址、绝对地址、符号地址、局部变量、全局变量、输入类指令、输出类指令、中间指令、基本逻辑指令、应用指令、结果寄存器、功能块

一个 PLC 所拥有指令的全体称为该 PLC 的指令系统。指令系统代表着 PLC 的性能或功能。一般讲，指令系统丰富，指令类型多、条数多、功能强的 PLC，所能干的事也多，性能当然也就好。

早期可编程序控制器指令很少，如 OMRON 公司的 C20 机，才 27 条指令。而且指令的类型很少，功能也不强。后期的产品，如 CPM1A 机，尽管为小型机，就有 41 种、148 条指令。大型机更多，如 CV1000，多达 300 多条。近期的产品，如 CJ1/CJ1H 机，尽管为小型机，其指令将近千条。而且还有功能很强的指令，如文字、文件处理指令。

PLC 指令系统是基于硬件的，加上所用的语言又未强制标准化，所以，各厂家 PLC 的指令系统都不相同。即使是同一厂家，型号不同，指令系统也不完全相同。

在编程之前，必须弄清楚 PLC 的指令系统。不熟悉指令系统，等于不懂语法用不好语言一样，无法编程。

从广义上讲，厂家提供的系统函数块与功能块，也应算为指令系统的一个部分。如西门子 PLC 的功能块 FB41、FB42、FB43 用于实现 PID 算法，实质上它就是别的 PLC 的 PID 指令。OMRON 公司的新型机也提供这样的功能块。

为了加深对 PLC 指令的了解，以下先对 PLC 指令作分类分析，然后再对一些通用指令作简单介绍。而特殊指令，将在以后的章节中继续介绍。

1. 按指令的操作数分

（1）考虑操作数数量，有：

1）无操作数指令，如 END（程序结束）指令、NOP（空指令，不作任何操作），仅有操作码，无操作数。这类指令不多。

2）单操作数指令，如 LD（装载）指令，除了操作码（LD），还要有操作数（位地址）。

3）多操作数指令，如 MOV（传送）指令，除了操作码（MOV），还要有被传送字源地址及目标地址。执行它后，则把字源地址的内容，传送到目标地址中去。

多操作数，有的操作数可多达 3 个。如 ADD（加）指令，在操作码 ADD 之后有 3 个地址。第一操作数为被加数；第二操作数为加数；第三个操作数为和。

指令在内存中占用的字与指令长度有关。单字的占一个字。多字的不只占一个字。CJ1 机指令在内存中占用的字与指令拥有的步（Step）有关。而实质上讲，CJ1 机的步与字基本上相同，OMRON 公司提供有换算方法。

提示：这里的操作数多少，与在指令执行中，参与操作的实际数的数量，并不是一回事。可能操作数只有两个，但实际参与操作的数可能是几十、几百，甚至上千。 真正参与

操作的数到底多少，是由指令的功能及特点决定的。

（2）考虑操作数本身特点，有：

1）位（bit）操作数，它相当于输入、输出点，或内部继电器。

2）数位（digit）操作数含 4 个位。作数位处理时，要用到它。

3）字节（Byte）操作数，含有 8 个位。作字节处理时，要用到它。

4）字（word）操作数，它含两个字节，16 个位。多数 PLC 都有这种操作数，特别是使用数据运算、处理时，更是这样。

5）双字（Double word）操作数，它含两个字（通道），可表示 8 位 BCD 码，或 8 位十六进制数。

6）多字操作数，有多个数据参与操作。

（3）考虑操作数地址特点，有：

1）即时数，即常数。可为 BCD 码（数码前加“#”），也可为十六进制码（数码前加“&”）。

2）直接地址。若为常数，则指令对这个常数进行操作；若为地址，则对这个地址的内容进行操作。

3）间接地址。以这个地址的内容作为地址，再用这个地址的地址的内容进行操作。如间接地址用在 DM 区，则地址代号前加“＊”（以 BCD 数计算抵制），或“&”（以十六进制数计算地址）号。

举例：若为＊DM0000，而 DM0000 的内容为 100（BCD 数），则指令要用 DM100 的内容进行操作。

如 &DM0000，而 DM0000 的内容为 100（十六进制数），则指令要用 DM256 的内容进行操作。

对具有索引寄存器及数据寄存器的 PLC，还可用索引寄存器及数据寄存器表示间接地址。

用索引寄存器间接寻址，可以是无偏移寻址。它把 IR@（@ 为索引寄存器的不同编号）的内容作为操作数的地址。使用时，在索引寄存器前加一逗号，以表示间接寻址。并根据指令或操作数，决定所指定的是位，还是字。如 LD，IR0，即为取 IR0 内容指出 I/O 内存地址上位的状态。再如 MOV #0001，IR1，即为把#0001 传送到 IR1 内容指出的地址上的字中。

用索引寄存器还可偏移寻址。偏移值可在 －2048 ~ ＋2047 间变化。

偏移值可以是常数，如执行 LD＋5，IR0 指令，即为把 5 与 IR0 内容相加，以此“和”指向的 I/O 内存作为地址，取该地址所指的位的状态。再如执行 MOV #0001＋31，IR1 指令，则是把 31 与 IR1 内容相加，以此“和”指向的 I/O 内存作为地址，把#0001 传送到该地址的字中。

偏移值也可以是存于 DR 数据寄存器中，它与 IR@ 中的内容相加，才是实际操作数的地址。如执行 LD　DR0，IR0 指令，即为把 DR0 的内容与 IR0 的内容相加后的内容作为指向 I/O 内存的地址，取该位地址的状态。再如，执行 MOV　#0001 DR0，IR1 指令，即为把 DR0 内容与 IR1 中的内容相加后的值作为 I/O 地址，把#0001 传送到该地址的字中。

偏移值还可以是自动递增或递减的。即在 IR@ 读 I/O 内存地址后或前，索引寄存器的内容增加 1、2 或减 1、2。如递增 1，其符号为“，IR@ ＋”。如递增 2，其符号为“，

IR@ + +”。如递减 1，其符号为“，－IR@”。如递减 2，其符号为“，－－IR@”。

如执行 LD，IR0 + +指令，即为取 IR0 内容指向的 I/O 内存地址上位的状态，然后再使寄存器的内容增 2。再如执行 MOV #0001，IR1 +指令，即为把#0001 传送到 IR1 内容指向的 I/O 内存地址上的字中，然后再使寄存器增 1。

再如执行 LD，－－IR0 指令，即为先使 IR0 的内容减 2，然后取 IR0 内容指向的 I/O 内存地址上位的状态。再如执行 MOV #0001，－IR1 指令，即为先使 IR1 的内容减 1，然后把#0001传送到 IR1 内容指向的 I/O 内存地址的字中。

提示：索引寄存器及数据寄存器使用，逗号“，”是不可缺少的。DR 字长为 16 位，而 IR 为 32 位。

（4）考虑操作数地址的表达方法，则有：绝对地址、符号地址。

1）绝对地址，它使用 PLC 厂家定义的地址。如 D0600，即 DM 区第 0600 字。

2）符号地址，用编程软件符号编辑器编辑的，与 PLC 厂家定义的地址对应的符号（用有意义的文字符号表达）地址。

提示：使用符号地址不仅可提高 PLC 程序的可读性，而且，还可使程序便于修改，便于重用。 是 PLC 编程技术重大进步。

作为符号变量操作数地址，如考虑其作用范围，则有：全局变量、局部变量。

（a）全局变量，它在 PLC 中定义。用于 PLC 所有程序。

（b）局部变量，它在程序中定义。只在本程序有效。

2. 按作用分

（1）输入类指令，用以处理输入信号及反馈信号，以建立逻辑条件。执行这类指令不产生输出，但它为输出类指令工作提供条件。显然，一个有效程序，不可能仅使用这类指令。

输入类指令有 3 种执行方式：

1）正常执行。每一扫描周期，都依它的操作数正常 I/O 刷新后得到的值，进行逻辑处理。

2）立即输入刷新执行。每次执行它前，先进行输入刷新，然后再依刷新后操作数取得的新值，进行逻辑处理。这样使用指令，要在它的代码之前加感叹号“！”。在梯形图上的符号为 ─┤!├─ 。

3）微分执行。有上沿微分与下沿微分。

上沿微分，当它的操作数从 OFF 到 ON 的那个周期，此操作数按 ON 处理，其它的均认为 OFF。这样使用指令，要在它的代码之前加向上箭头符号。在梯形图上的符号为─┤↑├─ 。

下沿微分，当它的操作数从 ON 到 OFF 的那个周期，此操作数按 ON 处理，其它的均认为 OFF。这样使用指令，要在它的代码之前加向下箭头符号。在梯形图上的符号为─┤↓├─。

输入指令执行方式与 PLC 的型号有关，不是所有 PLC 都有这么多的执行方式。

（2）输出类指令，用以产生输出，或执行某种处理。但是，产生什么输出，以及是否进行处理，要执行输入指令所建立的逻辑条件决定。所以，在执行这些指令之前，一般讲，总要先执行输入类指令。不然，输出怎么能去反映输入呢！

OMRON 公司的输出类指令也有 3 种执行方式：

1）正常执行。每一扫描周期均依执行它时的逻辑条件情况，处理该指令；到了输出刷新时，才把这个输出传送给输出锁存器。

2）立即刷新执行（Immediate Refresh，IR），处理该指令后立即进行输出刷新，把输出位的结果值送给相应的输出锁存器。这样使用指令，要在它的代码之前加感叹号“!”。

3）微分执行。有上沿微分与下沿微分。

上沿微分执行（Differentiated Up，DU），当它的执行条件从 OFF 到 ON 的那个周期执行，否则，即输入条件不变，或 OFF 或 ON，都不执行。这样使用指令，要在它的代码之前加符号“@”。

下沿微分执行（Differentiated Down，DD），当它的执行条件从 ON 到 OFF 的那个周期执行，否则，即输入条件不变，或 OFF 或 ON，都不执行。这样使用指令，要在它的代码之前加符号“%”。

输出指令的执行方式与 PLC 的生产公司及 PLC 的型号有关，不是所有 PLC 都有这么多的执行方式。

（3）中间指令，是为了简化编程、提高程序效率而新增的指令类型，老式的 PLC 多没有它。这类指令承上启下，既按在本指令之前建立的逻辑条件，执行本指令；又依本指令的执行情况，再建立逻辑条件，为后续指令的执行提供前提。

中间指令之后还可继以输入指令，然后才为输出类指令。以至于输入、中间指令多次相间，最后才为输出类指令。

如图 2-21 所示，这里用了 LD、两个比较及 AND 指令。这两个比较就是中间指令。这里，如果 0.01 ON，D100 大于等于 2.0，D200 大于等于 3.0，0.02 ON，则 20.01 ON，同时进行 D1000、D2000 浮点加运算，结果存入 D2006 中。否则，20.01 OFF，不进行浮点加运算。梯形图这样表达很简练，效率是很高的。只是，这样的梯形图就不大像电气原理图，与创立梯形图的初衷略有违背。也许这也算“与时俱进”吧。

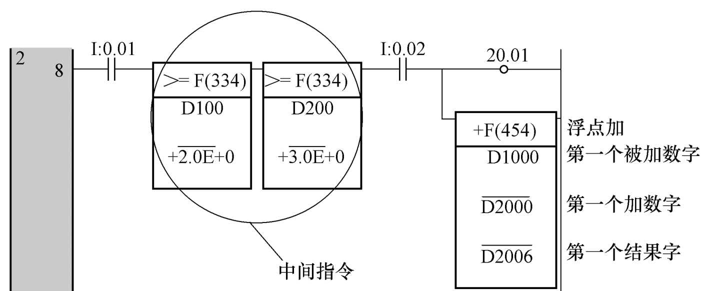


图 2-21　多个比较指令使用

这种指令的处理又称“EN”、“ENO”机制。即：每个指令都有“EN”，即输入条件，输入条件成立，才执行本指令；还都有“ENO”，执行结果，指令成功执行了，或执行后取得某期望的结果，则 ENO 为 1，否则为 0。

3. 按执行分

PLC 的输入指令，在每扫描周期中，总是执行的。多数输出指令，在执行（即与其有关的输入）条件具备时，也总是执行的，并立即产生执行后的效果。传统 PLC 的输出指令

也都是这样的。

但在新型的 PLC 中，有的指令就不完全是这样的。

如 PID 指令，尽管执行条件具备，设定又无不当之处，但它的执行周期不是取决于扫描周期，而主要取决于对 PID 工作周期的设定。

再如 AVG（求平均数）、SUM（求总数）这样表处理或文字处理指令，要在一个扫描周期内实现它的功能，所用的处理时间很长。新机型允许其分开在若干扫描周期内完成。这样可避免出现，执行这类指令时扫描周期过长，不执行时又较短的情况，从而可确保 I/O 响应时间的一致性。

4. 按使用分

（1）基本逻辑（有的称顺序）指令。用得最多，简易编程器上多有其对应的专用键。主要用于逻辑操作。

（2）应用指令，有的称为功能指令，可实现比逻辑操作更为复杂的功能。在简易编程器上，一般无与其对应的专用键。用简易编程器，输入这种指令有两种办法：一是用先输入功能键（FUN），后输入功能号，OMRON 及三菱有的 PLC 就是用这种方法；另一为在编程器上显示指令菜单，在菜单中选择所要输入的指令，西门子、松下公司的 PLC 就是用这种办法。

随着 PLC 技术的发展，功能指令越来越多。OMRON CV、CS1、CJ1 机，其功能码已超过两位数，如 END 指令，过去为 FUN（01），而现在为 FUN（001）。

而有的中、小 PLC，如 C200HS、C200Hα 机及 CQM1 机，功能指令也已是 100 多个，但多的不太多。OMRON 公司仍用两位数的功能码。两位数要区别 100 多个功能指令，怎么办？办法是：把功能指令分为两种：一为有固定的功能码，如 01，固定代表 END 指令；另一为无固定功能码，如 PULS（脉冲）指令，就没有固定的功能码，使用前现指定。没有固定的功能码的指令，OMRON 公司称之为扩展指令。

如 CQM1 机的 32 个扩展指令，出厂时，工厂已对其中的 17 个指定了功能码，不另作设定就可以用。其余的 15 个没有功能码，不能使用。若要使用，就得另改设定。改设定可使用简易编程器，也可使用编程软件。而且，这个设定要与用户程序一起存储，一起下载，一起使用。否则 PLC 无法认识这些指令，因而程序也不能正确执行。

提示：下载扩展指令设定前，应使 PLC 处编程模式，有的 PLC，如 CPM2A，其 DM6602（系统设定字之一）的高字节还要设为 1（也要使 PLC 处编程模式才能写这个字）。否则这个设定的下载是不可能的。

5. 按功能分

类型很多。以 CJ1 机为例，分有 32 类之多。对其再作些归类，还有：

（1）基本逻辑类指令，用于逻辑关系处理，是最常用、最基本的指令。

（2）定时、计数类指令，用于定时，或计数，也是经常要用到的指令。

（3）数据处理类指令，用于数据运算、传送、比较、译码、编码、移位及其它处理。

（4）流程控制类指令，用于控制程序执行流程。

（5）监控类指令，用于处理 PLC 或被控制对象的故障检测。

（6）处理 I/O 类指令，用于处理 PLC 应急 I/O 刷新或数据（信息）的入或出。

（7）通信类指令，用于处理 PLC 与 PLC，或 PLC 与计算机，或 PLC 与智能设备间通信。

（8）内存管理指令，用以管理 PLC 的存储区，存储卡。

以下将对其中若干类指令进行介绍，其它的将在介绍有关的内容时，再作说明。

2.3.1 基本逻辑操作指令

对二进制数（bit、位）进行逻辑操作，是 PLC 最简单的、也是最基本的功能。其所用的指令称为基本逻辑操作指令。所有的 PLC 都有这类指令。

这类指令可分为：读（输入类）与写（输出类）两种。

读指令是读操作数的逻辑值，并与已有的结果值进行逻辑运算，进而修改结果值。

写指令指的是把结果值写给操作数。

这里的“结果值”就是将要讨论的 R 寄存器的值，有的称为 RLO（Result of Logic Operation），即逻辑运算结果。西门子公司的 S7-200 称之为逻辑栈顶（The Top of the Logic Stack，TOS）。

1. 基本指令要点

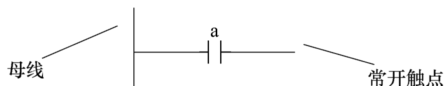
（1）LD 指令，为装载指令。有的 PLC 用 ST（起始）作为它的符号。

LD 指令的作用是，把指定操作数（位）的值（0 或 1，分别代表断或通，工作或不工作……）送入结果寄存器 R，并把结果寄存器的原有内容送入堆栈 P（有的为第二个 RLO，不是堆栈）。

LD 指令的语句表符号格式为

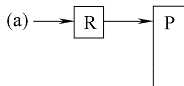
指令地址　LD　操作位地址

梯形图符号为



它为常开触点，并总是与梯形图的母线相连。

它的功能为



这里 a 为操作数的地址，括号代表 a 的值。R 为结果寄存器，P 为堆栈。堆栈为先进后出的存储单元，一般长度为 8 位（bit），与 PLC 型别有关，CJ1 机为 16 位。8 位时，可存储 8 个二进制数，再续存时，最先存的数丢失。堆栈主要在逻辑块操作，或需多个输入条件时用到它。

LD NOT 指令，它加一个 NOT，表示把操作数的值取反，然后再送结果寄存器。其它的同 LD 指令。在梯形图上，它用常闭触点代表，在两短平行线的基础上，再加一小斜线。

有的 PLC，LD 及 LD NOT 指令还可加感叹号（!）及上或下箭头（↑↓），其含义前已介绍。加了这个感叹号、向上及向下箭头，使指令的功能大为增强，一个指令可起到多个指令的作用。

S7-200 逻辑栈，除了栈顶，还有 8 位栈体，也可暂存 8 位（bit）。它的栈体相当于这里

的栈，而栈顶则相当于这里的结果寄存器。

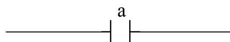
（2）AND 指令，为与操作指令。有的 PLC 的符号为 A 或别的符号，但作用相同。

AND 指令的作用是把操作位的值与 R 的值相与，然后再送入 R 中。这时，堆栈的内容无变化。

其语句表的符号为

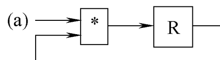
指令地址　AND　操作位地址

梯形图符号为



为常开触点，代表串联。

其功能为



这里 a 为操作数的地址，括号代表 a 的值。R 为结果寄存器。

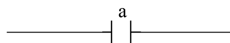
AND NOT 指令。它也是加一个 NOT，表示把操作位的值取反，然后再与结果寄存器的值相与。其它的同 AND 指令。它在梯形图上为常闭触点。

有的 PLC　AND 及 AND NOT 指令也可加感叹号、上下箭头。含义同 LD 指令。

（3）OR 指令，为或操作指令。有的 PLC 用 O 或别的符号表示。

OR 指令的作用是把操作位的内容与 R 中的内容相或，然后再存入 R 中。这时，堆栈的内容无变化。

梯形图符号为

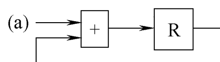


OR 指令的语句表符号为

指令地址　OR　操作位地址

为常开触点，代表并联。

其功能为



这里用的符号同 AND 指令。

OR NOT 指令，加一个 NOT，代表先取反而后才或。它在梯形图上为常闭触点。

有的 PLC，其 OR、OR NOT 指令也可加感叹号，上下箭头，含义同 LD 指令。

（4）AND-LD、OR-LD 指令，为块与、块或指令，无操作数。有的 PLC 用别的符号。

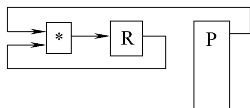
其作用是把结果寄存器的内容与堆栈的内容作逻辑与，或逻辑或，然后再送结果寄存器。

其语句表符号为

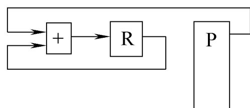
指令地址　AND-LD 或指令地址 OR-LD

它在梯形图上代表两组触点的串联或并联。

AND-LD 功能为



OR-LD 功能为



AND-LD、OR-LD 指令用于触点组间的串联或并联，是很有用的指令。如图 2-22 所示，其对应的助记符指令也已列出。

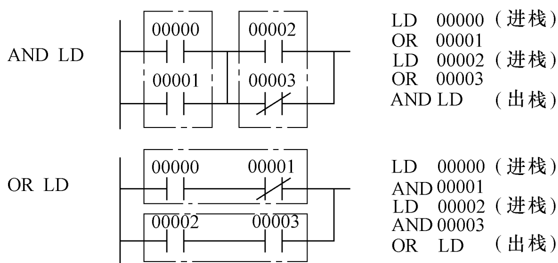
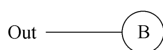


图 2-22　触点组间的串联或并联

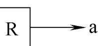
（5）OUT 指令，为写指令。语句表的符号为

指令地址　OUT　操作位地址

梯形图符号为输出线圈，可用圆圈或方块表示。如下图所示。

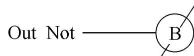


其功能为



这里 a 为操作数的地址。执行 OUT 指令后堆栈内容不变，R 的内容也不变，只是把 R 的内容传给 a。

OMRON 公司的 OUT 指令还可加 NOT。如下图所示。



其含义是把 R 先取反，然后再传给 a。表示符号为在 OUT 的符号基础上，加一斜线。

有的 PLC，OUT 及 OUT NOT 指令也可加感叹号（!），如 CJ1 机。含义同上。

（6）S、R 指令，置位、复位指令。其操作数为位地址，也是一种输出指令。

S 为置位（SET），R 为复位（RESET）。只要执行本指令时，结果寄存器 R 的内容为 1，则执行。否则不执行，其操作数值不变。执行置位（SET）指令，其操作数置 1。执行复位 RESET）指令，其操作数置 0。

如图 2-23 所示，若 0.00 ON，0.01 OFF，则 10.00 置 1。若 0.01 ON，则 10.00 置 0（因它在 SET 指令之后执行，可更改 SET 指令执行的结果）。若 0.00 及 0.01 OFF，则 10.00 值不变。

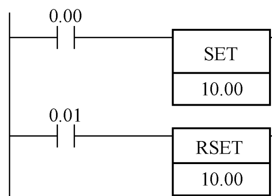


图 2-23　置位、复位指令使用

把这两者复合在一起，为 KEEP 指令。类似于 R_S 触发器。有两个输入端，一为 R 端，另一为 S 端，分别对操作位置 0（复位）或置 1（置位）。

S、R 指令可分开置于程序的不同位置，用起来较灵活。而 KEEP 指令则要依此执行这两个指令，先 S 后 R。

这 3 个指令助记符的符号为

指令地址　S　操作位地址

指令地址　R　操作位地址

指令地址　KEEP　操作位地址

这 3 个指令都是输出指令。使用前都要对结果寄存器 R 赋值。

S、R 指令前各赋一次值，KEEP 指令之前要赋两次值（要两次使用 LD 指令）。在梯形图上的表示为线圈，或方块。S、R 仅一个入端，而 KEEP 要有两个入端。如图 2-24 所示。

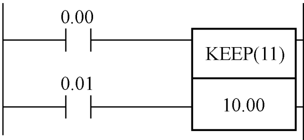
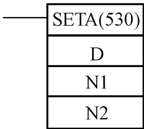


图 2-24　KEEP 指令使用

提示：图 2-23 与图 2-24 两个程序，表面上功能是相同的，但实际是有区别的。如果用 10.00 代替 0.01，当 0.00 ON 时，图 2-23 程序可使 10.00 ON、OFF 按扫描周期交替出现，而图 2-24 程序 10.00 永远不可能 ON。图 2-23 程序的这个特点，可用以实现“单按钮起、保、停”控制。

CJ1 机有，SETA 指令，其功能码为 530。可对多个位操作。梯形图符号如下：



这里，D 为字的地址；

N1 为操作数的起始位；

N2 为要置位的操作位个数，可用常数，也可用地址。

图 2-25 所示为使用 SETA 指令的例子。

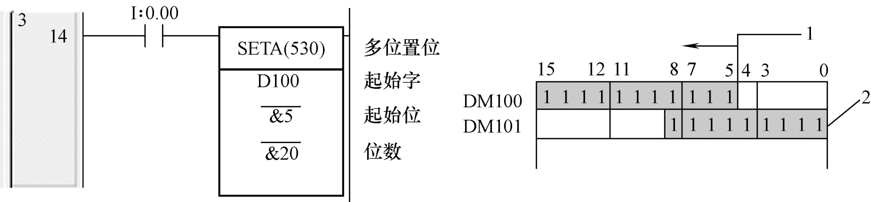
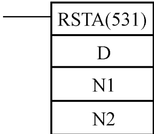


图 2-25　SETA 指令使用例子

1—N1：第 5 位　2—N2：20 位

当 0.00 ON，将使 D100 的第 5 位（N1 = 5）开始，直到 DM101 的第 8 位，共 20（N2 = 20）位置 1。

多个位复位的是，RSTA 指令，其功能码为 531。梯形图符号如下：



这里，D 为操作位所在的字地址；

N1 为操作位的起始位；

N2 为要复位的操作位个数，可用常数，也可用地址。

图 2-26 所示为使用 RSTA 指令的例子。依此例，当 00000 ON 时，将使 DM0100 的第 3 位开始到 DM0101 的第 6 位，共 20 位置 0。

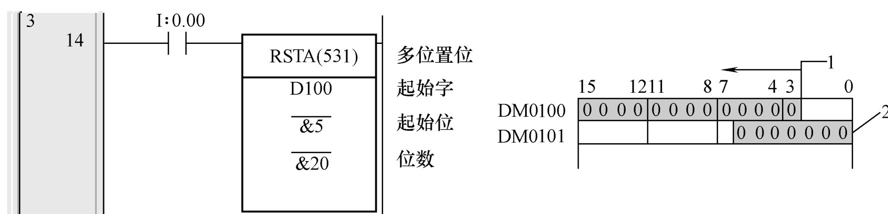


图 2-26　RSTA 指令使用例子
1—N1：第 3 位　2—N2：20 位

提示：SETA 指令及 RSTA 指令不仅可实现多位操作，而且，可对 DM 的位也可进行逻辑操作。

（7）微分指令，DIFU 为上沿微分，DIFD 为下沿微分。

它的操作数也是位地址。当执行 DIFU 时，R 的内容从 OFF（0）变为 ON（1），则操作数的内容为 1（ON）一个扫描周期。

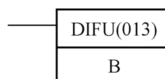
当执行 DIFD 时，情况与 DIFU 相反。R 从 ON 变到 OFF，操作数 ON 一个扫描周期。

有的 PLC，在一个程序中，微分指令的使用次数是有限制的，如 OMRON C 系列 P 型机，最多只能使用 48 次。

其助记符指令为

指令地址　DIFU　B

其梯形图指令为



这里 B 为输出位，它的功能码为 013。在逻辑条件从 OFF 到 ON 时，它使输出位 ON 一个扫描周期，如图 2-27 所示。

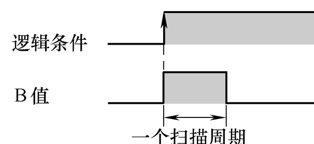
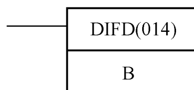


图 2-27　上沿微分指令示意图

其助记符指令为

指令地址　DIFD　B

其梯形图指令为



这里 B 为输出位，它的功能码为 014。在逻辑条件从 ON 到 OFF 时，它使输出位 ON 一个扫描周期，如图 2-28 所示。

有的 PLC 的微分指令不作为输出指令，而作为中间指令。它可加在一组输入指令之后，加上它，然后再送给输出指令，用起来也很方便。CJ1 机就有此指令，如 UP、DOWN 指令。

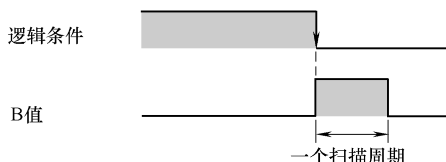
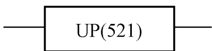


图 2-28　下沿微分指令示意图

UP 指令梯形图符号如下：



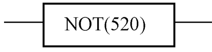
它的功能号为 521。执行它时，可对其输入作上沿微分。

DOWN 指令梯形图符号如下：



它的功能号为 522。执行它时，可对其输入作下沿微分。

此外，还有 NOT 指令，其梯形图符号如下：



它的功能号为 520。执行它时，可对结果寄存器取反。

2. 基本逻辑指令应用

有了基本逻辑处理指令，PLC 也就有了“与、或、非”逻辑的处理功能。在理论上讲，这就足以完成所有控制与计算，但最常用的还是用于控制输出。

（1）等效控制输出

输出仅取决于控制输入现状态。这是用开关控制电器工作的特点，也是组合逻辑的特征。图 2-29 所示的就是这类输出。它把开关接输入点 0.00，负载用灯接输出点 10.00，然后运行图示程序。

图 2-29 所示用户程序分别用 LD、IL 表示。从图中记载的各周期 I/O 状态可知，在周期 1 ~ n 之前，由于开关已合上，输入点 0.00 的值为 1。经运行程序后，输出点 10.00 也变为 1。进而输出点合上，输出电路通，灯亮。在周期 n 及以后，开关断开，则输入点 0.00 的值为 0，输出点 10.00 也变为 0，输出点断开，输出电路断，灯灭。可知，这里灯的工作惟一取决于开关的状态。

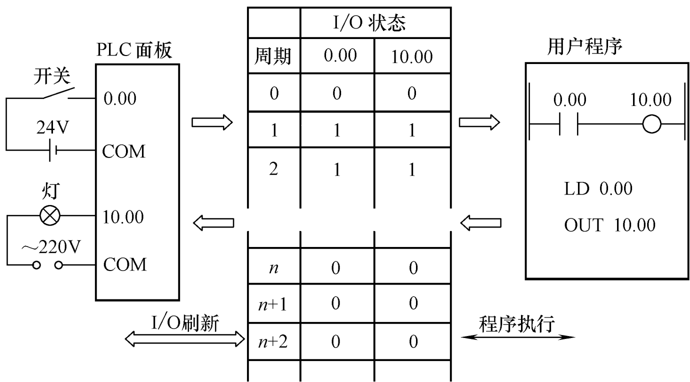


图 2-29 等效控制输出实例

（2）长效控制输出

控制输出不完全取决于输入的现状态，还与输入的历史有关。输入对输出有长效作用。这是继电器控制电器工作的特点，也是时序逻辑的特征。如图 2-30 所示，这里用两个按钮控制接触器，再用接触器控制一盏灯。

该图用户程序分别用 LD、IL 表示。此程序已在本书第 1 章就说明过，在此不再赘述。这里，当 SB1 合、或 SB2 合都可产生长效输出。

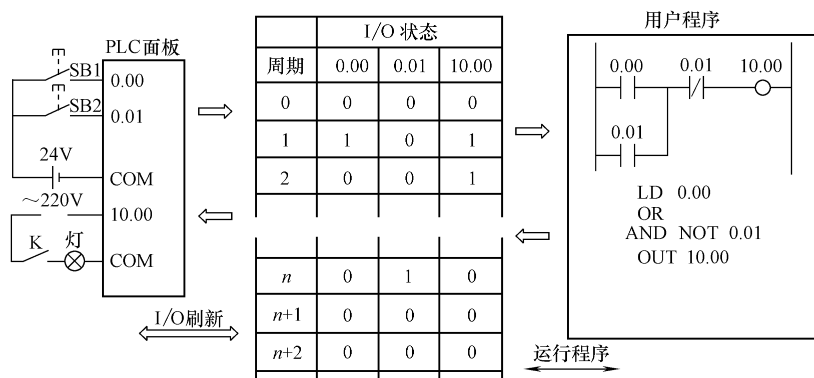


图 2-30　长效控制输出实例

长效输出的特点是存在输出对输入的反馈。

提示：也可以使用置位、复位指令实现长效输出，效果相同。

（3）短效控制输出

它的控制输出也不完全取决于输入的现状态。其输入对输出仅有短暂的作用，故称短效输出。如图 2-31 所示的梯形图程序，用开关控制一盏灯，但中间插入内部辅助继电器 200.00。

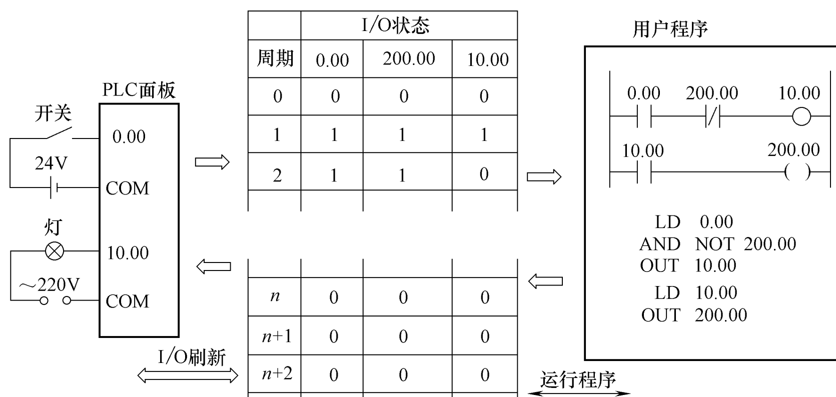


图 2-31　短效控制输出实例

从分析该图可知，用户程序分别用 LD、IL 表示。分析可知，当开关合上，10.00 则产生一个周期的输出。而当开关断开，则什么输出也没有。

所以，在稳定状态下，10.00 永远为 0。但在过渡状态下，它可为 1。像这样仅为一个周期的信号也称脉冲信号，或微分输出。用其去控制灯的工作当然没有意义。但这在内部逻辑处理时，它是很有用的。

这里是当 0.00 从 OFF 到 ON 时，10.00 ON 一个扫描周期，也称为微分上升沿输出。有时需要微分下降沿输出，图 2-32 所示，就可以实现这样的输出。只是，这里把 0.00 从常开改为常闭。同时增加条 3，并在条 1 串入触点 200.01。这样做的目的是，在 PLC 开始

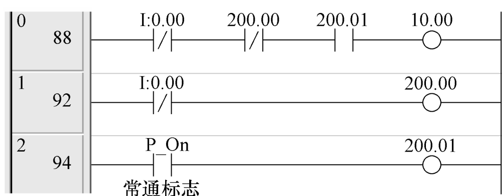


图 2-32　微分下降沿输出

运行第一周期不产生脉冲。只有当 0.00 先 ON，后 OFF 时，才产生微分下降沿的脉冲输出。

2.3.2 定时、计数指令

定时与计数指令主要用于定时与计数。本质上也是一种逻辑输出指令。只是，它是延时实现，或计数满后实现。所以，有的 PLC，如三菱公司 FX 机，起用定时器、计数器就是用 OUT 指令，只是其操作数用定时器、计数器，并在使用它时，同时对定时值、计数值也作设定。

定时与计数指令使用的工作单元，为定时器与计数器。如 TIM 1，表示定时器 1，既可把它看成标志位，又可把它看成是字（其内容与现值有关）。有的 PLC 对这两者的表示是分开的。

1. 普通定时指令（TIM）

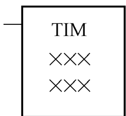
指令的助记符为

地址　TIM　定时器编号

　　　　设定值（即时数或地址）

在这指令之前，当然要对寄存 R 赋值，即写 R，建立条件，或说连一个输入端。

指令在梯形图上的符号是方框或圆圈。如方框为



这里×××为指定定时器的编号及设定值。

普通定时指令的设定值的单位为 0.1s，仅能区别 100ms 的精度。设定值、现值都用 BCD 码表达。最大设定值可达 9999，即 999.9s。有的 PLC 也可十六进制数，最大设定值可达 65535。

图 2-33 为 OMRON PLC 使用定时器的梯形图。其程序为

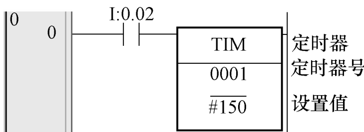
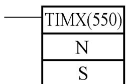


图 2-33　定时器使用

这里 0001 为定时器号，#150 为定时设定值，单位为 0.1s，故定时值为 15s。0.02 ON 后延时 15s，TIM0001 常开触点 ON。而一旦 0.02 OFF，则 TIM 0001 立即 OFF。

OMRON CJ1H 机还有 TIMX 指令。其功能与 TIM 相同，只是它是按十六进制计时。其格式为



这里，N 为定时器编号；

　　　S 为定时设定值，按十六进制计；

　　　550 是它的功能号。

提示：虽然这种 PLC 同时有 TIM 及 TIMX 指令，但在编程前要用编程软件，在 PLC 属性栏中，先作选择，而且，只能选用其中的一种。默认选定为 TIM。 其它带“X”的指

令也都有此情况。

2. 高速定时指令（TIMH）

与普通的定时指令无重大区别，只是它的计数单位为 0.01s，而不是 0.1s。有的高速计数器的单位为 0.01s，可实现 10ms 级的计数。

TIMH 不太常用，OMRON 指定其为功能指令。指定其功能码为 15，CJ1 机为 015。高速定时指令所用的定时器编号有限制，一般用前边的号（如 000 ~ 014）。这些定时器可中断工作，可保证能区分较小的定时单位。

CJ1 机还有高速定时指令 TIMHH，它的定时值设定单位为毫秒。可知，它可处理毫秒级事件。这也是过去 OMRON PLC 所没有的。它的功能码为 540。

此外，CJ1H 机还有与 TIMH、TIMHH 对应的 TIMHX、TIMHHX 指令，所不同的也只是它们是按十六进制数计时，定时值设定范围可增大近 6 倍。

3. 其它定时指令

（1）累计定时指令（TTIM）：CV1000，C200α 机开始有此指令，CJ1 机也有此指令，用以累计计时。它是增计时，计时单位为 0.1s。输入端 ON 时计时，OFF 不计时，但不复位。再 ON，再计，并累计计时，直到达到设定值，计时停止，并产生输出。计时器的复位用复位位 ON。

CJ1 机也还有与 TTIM 对应的 TTIMX 指令，所不同的也只是它是按十六进制数计时。

（2）8 位计时指令（TIML）：CV1000 机开始有此指令，是普通定时指令的加长，设定值可达 8 位，即 99999999。可计 115 天。

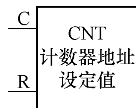
CJ1 机还有与 TIML 对应的 TIMLX 指令，所不同的也只是它是按十六进制数计时。最多可计 49710 天。

（3）多输出计时指令（MTIM）：CV1000 机开始有此指令，可产生 8 个输出。这 8 个输出相应于 8 个设定值。计时时，计时是增计数，现值不断增大，与某一设定值相比，大或等于后者时，即产生相应输出。这个定时指令，一个相当于多个，扩大了定时器的功能。

CJ1H 机也还有与 MTIM 对应的 MTIMX 指令，也是用十六进制计时。

4. 减计数指令（CNT）

它实现减计数。有两个输入端，一为计数端，另一为复位端。指令的梯形图格式为



它的工作情况是：复位端（R）的逻辑条件为 ON，停止计数，现值复位为设定值。复位端 OFF，允许计数。这种情况下，当计数端（C）的逻辑条件从 OFF 到 ON 时，在该扫描周期，计数器的现值减 1。其它情况下，现值不变。当现值减为 0 时，产生输出，且现值保持为 0。

CJ1H 机还有 CNTX 指令，所不同的它用十六进制计数。所以，它的计数范围可扩大到 65535。

5. 可逆计数指令（CNTR）

它为功能指令，功能码为 12，CJ1 机为 012。除了有复位端，还有两个计数端，一个为正计数端（U），一个为减计数端（D）。其梯形图格式为

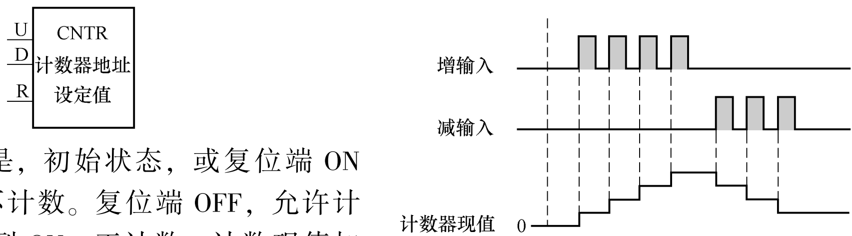


图 2-34　可逆计数示意图

其工作情况是，初始状态，或复位端 ON 时，现值为 0，不计数。复位端 OFF，允许计数。正端从 OFF 到 ON，正计数，计数现值加 1；负端从 OFF 到 ON，减计数，计数现值减 1。具体计数情况如图 2-34 所示。

计数增到设定值，再增计 1 个数，则现值变为 0，且产生输出，计数完成标志位 ON，如图 2-35 所示。

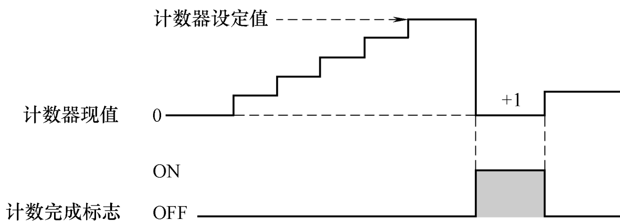


图 2-35　计数增到设定值再增 1 计数

计数到现值为 0，再减 1 个数，则现值变为设定值，也产生输出，计数完成标志位 ON，如图 2-36 所示。

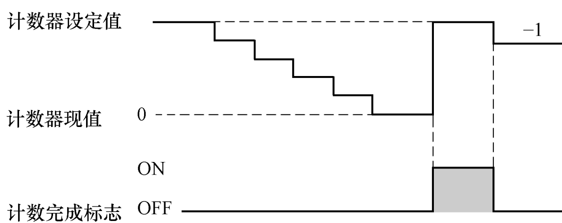
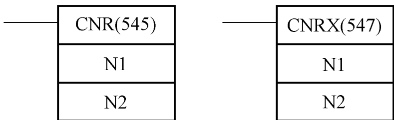


图 2-36　计数减到 0，再减 1 计数

CJ1H 机还有 CNTRX 指令，与 CNTR 不同的是，它用十六进制，而不是用 BCD 码计数。所以，它的计数范围可扩大到 65535。

6. 复位定时器/计数指令（CNR、CNRX）

它用于成批复位定时器及计数器。CNR 用于 BCD 数定时器、计数器，而 CNRX 则用于十六进制数定时器、计数器。它们的格式为



这里，N1 为将成批复位的定时器、计数器的开始编号；

N2 为将成批复位的定时器、计数器的结束编号；

545、547 是本指令的功能码。

再次提示：OMRON 公司的 CJ1H 等机型，虽然支持十六进制计数、定时的指令，但在一个程序中，不能与 BCD 码计数的计数、定时指令同时使用。到底用那种计时、定时指令，可用 CX-Programmer 软件（见后，本章第 6 节）设定。

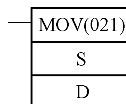
2.3.3　数据处理指令

数据处理指令很多，占 PLC 指令集的相当大部分。大都以字、多字为单位操作。具体有：传送指令、比较指令、移位指令、译码指令、各种运算指令、文字处理指令，等等。

1. 传送指令

最简单、最常用的传送指令为 MOV（021），021 是它的功能号。

MOV（021）是字传送指令，执行之后（执行它时，若逻辑条件具备，即此时的结果寄存器 R 的内容为 1，则执行），把源地址的内容或某即时数传送到某目标地址。传送后，源地址内容不变。其梯形图格式为



这里，S 为源地址，也可是即时数；

D 为目标地址。

S、D 的内容可以是 BCD 码，或十六进制码。S、D 可以为直接地址，也可为间接地址，即该内容仍是地址，后者的内容才是要传的数。

传送指令也影响标志位（25506 特殊继电器，它反映相等的特点），传送的数为 0 时，置其为 1，不然置 0。

还有 MNV（022）指令，它与 MOV 不同的只是传送之前，先把要传的内容取反，然后再传。高档 PLC 还有双字传送指令，MOVL、MNVL。

此外，还有多种传送指令。有：

多字传送指令 XFER（070）。也称块传送指令，可把若干连续地址的内容分别传送给对应的连续的目标地址。只要设好要传的数据的起始地址，目标的起始地址及要传的字数就可以了。

块设定指令 BSET（071）指令。它可把一个字的内容设定到指定的连续存储区中，只要指出该区的起始地址及末了地址。这个指令可很方便地用于对 PLC 的一些存储区进行初始化。

字交换指令 XCHG（073）。可进行两个地址内容的交换。高档的 PLC 还有双字交换指令 XCHGL。

带偏移目标地址的传送指令 DIST（80）。可把源地址的内容传送给某基址加偏移地址后的地址。这种传送比较灵活，便于存储数据，或从同一子程序中取出的数存于不同的单元中。带偏移源地址的传送指令，COLL（81）。可把某基址加偏移地址后的地址的内容传送到某个目标地址。这种传送便于取数，或同一子程序可使用不同的参数。

除了字、双字、多字传送，还有 BCD 码的位（digit）及十六进制的位（bit）传送，等等。这些指令给数据处理都提供了方便。表 2-14 ~ 表 2-17 列出了 CJ1 机的一些主要的传送指令。

表 2-14　传送指令 1

传送 MOV @MOV IMOV !@MOV 021	MOV(021) S D S: 源 D:目的	把一个字的数据传送到指定字中 源字 不改变位状态 目的字
双字传送 MOVL @MOVL 498	MOVL(498) S D S:源起始字 D:目的起始字	把两个字的数据传送到指定的字中 S　S+1 不改变位状态 D　D+1

表 2-15　传送指令 2

取反传送 MVN @MVN 022	MVN(022) S D S:源 D:目的	把一个字的数据的反码传送指定字 源字 位状态取反 目的字
双字取反传送 MVNL @MVNL 499	MVNL(499) S D S:源起始字 D:目的首字	把两个字的数据的反码传送到指定的字中 S　S+1 位状态取反 D　D+1
位传送 MOVB @MOVB 082	MOVB(082) S C D S:源字或数据 C:控制字 D:目的字	传送指定的位 C　m　n n S m D
多位传送 XFRB @XFRB 062	XFRB(062) C S D C:控制字 S:源起始字 D:目的首字	传送指定数目的连续位 15　8 7　4 3　0 C　n　m　l n l S m D

表 2-16　传送指令 3

数字传送 MOVD @ MOVD 083	MOVD(083) S C D S:源字或数据 C:控制字 D:目的字	传送指定的数字或多个数字(每个数字由4个位组成) 15 12 11 8 7 4 3 0 C D I n m n m S I D
块设置 BSET @ BSET 071	BSET(071) S St E S:源字 St:开始字 E:结束字	将相同字的内容复制到几个连续字中 源字 目的字 S St E
数据交换 XCHG @ XCHG 073	XCHG(073) E1 E2 E1:第一个交换字 E2:第二个交换字	把两个指定字的内容进行交换 E1 E2
双字数据交换 XCGL @ XCGL 562	XCGL(562) E1 E2 E1:第一个交换字 E2:第二个交换字	把一对连续字的内容与另一对连续字的内容进行交换 E1 E1+1 E2 E2+1

表 2-17　传送指令 4

单字分配 DIST @ DIST 080	DIST(080) S Bs Of S：源字 Bs：目的基地址 Of：偏移	把源字传送到目的字中，目的字的地址由目的基地址加上一偏移值 S Bs Of n n Bs+n
数据收集 COLL @ COLL 081	COLL(081) Bs Of D Bs：源基地址 Of：偏移 D：目的字	把源字(在源基地址上加一偏移值)传送到目的字 Bs Of n n Bs+n D

2. 比较指令

常用的比较指令为 CMP（020），功能码为 020。执行它时，实现两个数的比较，并根据比较结果使相应的标志位置位。

比较结果位有 3 个：

EQ（等于），第一、第二比较数相等，OMRON 公司以前机型是使 25506 ON；

LE（小于），第一个数小于第二个数，OMRON 公司以前机型是使 25007 ON；

GR（大于），第一个数大于第二个数，OMRON 公司以前机型是使 25505 ON。

提示：在比较指令与取得比较结果之间，不能如图 2-37所示那样，插入指令 B。因为执行指令 B，有时可能改变在比较时得到的结果。那样，A 得到的结果，就可能有误。

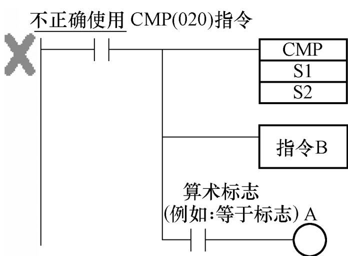


图 2-37　不正确使用比较指令示意图

此外，还有表比较指令 TCMP（085）：可把一个数与若干个数比较。与哪数相等，则指定字中相应位 ON。否则，OFF。表比较指令，要指定第一比较数，同时，还要指定数表（一般为 16 个数）及指定输出通道。表比较指令功能比 CMP 指令要强得多。

块（范围）比较指令 BCMP（－），较高档的 PLC 才有此指令。它把数与比较表中 16 对数比较，这 16 对列出上下限。这个数处于某上下限之间（含上、下限本身），指定字的相应位 ON。否则，OFF。

还有很多比较指令。特别要提到的是符号比较指令。它使用等于、大于及小于符号及其组合进行单字、双字比较。而且，指令还可作为中间指令使用。用起来很灵活。表 2-18 示的即为此类指令的简要介绍。

表 2-18　符号比较指令概要

指令 助记符 代码	符号/操作数	功　能	位置 执行条件
符号比较 (不带符号) LD,AND,OR + =, < >, <, < =, >, > = 300(=) 305(<>) 310(<) 315(<=) 320(>) 325(>=)	符号和选项 S_1 S_2 S_1：比较数据 1 S_2：比较数据 2	符号比较指令(不带符号)以 16 位二进制数据形式对两个数值(常数和/或指定字的内容)进行比较,并且当比较条件是真时,形成一个 ON 执行条件。有 3 种类型的符号比较指令:LD(LOAD),AND 和 OR LD：< S_1 S_2　当比较结果是真时，执行条件为 ON AND：< S_1 S_2　当比较结果是真时，执行条件为 ON OR：< S_1 S_2　当比较结果是真时,执行条件为ON	LD:不需要 AND,OR:需要

（续）

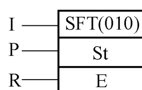
指令 助记符 代码	符号/操作数	功　　能	位置 执行条件
符号比较 （双字，不带符号） LD，AND，OR += ， <>，<，<=，>，>=+ L 301（=） 306（<>） 311（<） 316（<=） 321（>） 326（>=）	S_1：比较数据 1 S_2：比较数据 2	符号比较指令（双字，不带符号）以不带符号的 32 位二进制数据对两个数值（常数和/或指定字的内容）进行比较，并且当比较条件是真时，形成一个 ON 执行条件。有 3 种类型的符号比较指令：LD（LOAD），AND 和 OR	LD：不需要 AND，OR：需要
符号比较 （带符号） LD，AND，OR + = ， <>，<，<=，>，>= +S 302（=） 307（<>） 312（<） 317（<=） 322（>） 327（>=）	S_1：比较数据 1 S_2：比较数据 2	符号比较指令（带符号的）以带符号的 16 位二进制（4 位数十六进制）对两个数值（带数和/或指定字的内容）进行比较，并且当比较条件为真时，形成一个 ON 执行条件。有 3 种类型的符号比较指令：LD（LOAD），AND 和 OR	LD：不需要 AND，OR：需要
符号比较 （带符号双字） LD，AND，OR + = ， <>，<，<=，>，>= +SL 303（=） 308（<>） 313（<） 318（<=） 323（>） 328（>=）	S_1：比较数据 1 S_2：比较数据 2	符号比较指令（双字，带符号）以带符号的 32 位二进制数据（8 位数十六进制）对两个数值（常数和/或指定字的内容）进行比较，并且当比较条件是真时，形成一个 ON 执行条件，有 3 种类型的符号比较指令：LD（LOAD），AND 和 OR	LD：不需要 AND，OR：需要

提示：比较指令对应操作数的格式应一致，否则无法得到预期的结果。 而且，这里介绍的只是一种比较指令，实际还有很多，可根据数据类型选用。

比较指令常用于逻辑判断，是实现智能控制必不可少的指令。

3. 移位指令

最常用的为 SFT（010）指令，它的梯形图符号为



这里，I 为输入信号；P 为移位脉冲；R 为复位信号；St 为起始通道地址；E 为终止通道地址。

复位端 ON 时，从 St 到 E 的地址的内容全部置 0，并禁止移位。

R 端 OFF，允许移位。移位由移位脉冲实现。当 P 从 0 到 1 时，把 I 的内容（位，0 或 1）移入 St 通道的 00 位，而 00 位的原内容移给 01 位……，St 通道的 15 位移给 St + 1 通道的 00 位……，依此类推。E 的 15 位原数据丢失。

每产生一个脉冲，移一个位。

这里的 3 个端（I、P、R）用 3 个 LD 开头的输入指令产生，在语句表上必须为

××××　LD　××××

××××　LD　××××

××××　LD　××××

××××　SFT（010）

　　　St

　　　E

此外，还有多种多样的移位指令：

算术左移 ASL（025）：仅对一个字移位，执行一次本指令移一次位。移位时用 0 移入最低位（00 位）。原 00 位的内容，移入 01 位……，15 位的内容移入进位位 CY（25504）。而原进位位的内容丢失。

算术右移 ASR（026）：与 ASL 不同的只是它为右移，先把进位位的内容移入字的最高位（15 位），原 15 位的内容移入 14 位……，原 00 位的内容丢失。

循环左移 ROL（027）：它与 ASL 不同的只是它的 CY 的内容不丢失，要传给 00 位，以实现循环。

循环右移 ROR（028）：与 ASR 不同的是 00 的内容不丢失，传给进位位，以实现循环。还有可逆移位指令，由控制字去控制左，还是右移，并可实现多字移位。

除了二进制的位（bit）移位，还有数位（digit）移位，可左移 SLD（074），也可右移 SRD（075）。移位的对象可以多个字。

还有字移位 WSFT（016），以字为单位的移，执行一次本指令移一个字。移时 0000 移入起始地址（最小地址），起始地址的原内容移入相邻的较高地址，……最高地址（结束地址）的内容丢失。多次执行本指令，可对从起始到结束地址的内容清零。

4. 译码指令

用以译码，以适应数据使用或实现控制的需要。

最常用的为 BCD 码与 BIN 二进制码转换用指令。BCD（024）为二进制码转换成 BCD 码指令。BIN（023）为 BCD 码转换二进制码指令，还可用 BCDL 及 BINL 进行两字长转换。

此外，还有其它译码指令。

如：4 转 16 的 MLPX（076）指令：它可根据源地址十六进制数的一个数位（digit）的内容（0 ~ F），使相应通道的二进制位（bit）置成 1（其它非指定位置成 0）。如某位内容为 7，可使指定通道的 07 位（bit）置成 1，其余的均为 0。最多可用 4 个数位（digit）使 4 个通道的相应位置 1。到底有多少个数位，各数位对应于哪个通道由控制字确定。

还有与 MLPX 相反的译码指令 DMPX（077）。它为 16 到 4 译码，是前者的反过程，把

1～4 个通道中最高位（bit）为 1 的位号，分别置入指定通道的某个数位（digit）。有多少通道及通道与哪个指定字数位（digit）对应由控制字确定。

这两条指令的控制字高位设成 1，可进行 8 到 256 或 256 到 8 的译码。256 为 16 个字组成的 256 个位（bit）。8 为 8 个字节为一组，每组两位 16 进制数，其值变化范围为 00～FF，对应于 256 位（bit）的 000～255 位。只是 8 到 256，或 256 到 8 最多只能进行两组，不像 4 到 16 及 16 到 4 可进行 4 组。

此外还有：7 段译码指令 SDEC，可把 BCD 码译成 7 段码，用于数字显示。ASCII 码转换指令 ASC，把十六进制数译成 ASCII 码，还可加或不加奇或偶校验。十六进制数译码指令 HEX，把 ASCII 码译成十六进制数。小时到秒译码指令 SEC，把小时数译成秒数。秒到小时译码指令 HMS，把秒数译成小时数。列译成行译码 LINE，把 16 个字的指定同一个位（bit）的内容，变成一个指定字的内容。行列列译码 COLM，与 LINE 具有相反的过程。

等等。

5. 数据运算与处理指令

最常用的为一个字的 BCD 码 +、－、×、/指令。还有两个字（8 位 BCD 码）+、－、×、/。

还有 16 位及 32 位二进制的 +、－、×、/，CJ1 机还可带符号位的运算。

还有加 1，INC（038）及减 1，DEC（039）两个也是较常用的指令（CJ1 机不是用此符号）。

这些运算都要影响进位位，而且进位位也都参加运算。

为此，在运算之前要对进位位（CY）进行操作。有两个指令可做此工作。即：

STC（040）：置进位位，进位位置 1。

CLC（041）：清进位位，进位位置 0。

提示：进行 BCD 码加减运算时，进位位（CY）也是其运算成员。它既是运算原始数据（1 或 0）之一，又要存放运算结果（有进位或借位时置 1，否则置 0）。如加运算是 Au（被加数）+ Ad（加数）+ CY→CY，R（结果）。如减运算是 Mi（被减数）－Su（减数）－CY→CY，R。

此外，还有开方，浮点除。还有浮点 +、－……

还有，在一组数中求最小值（MIN），在一组数中求最大值（MAX），求一组数的平均值（AVG），求一组数的总和（SUM）。

还有可进行三角函数运算，有的还进行对数、指数运算，可进行一组数的数值插值运算。

随着 PLC 技术的发展，以及满足模拟量控制、脉冲量控制及通信的需要，综合运算指令越来越多，功能也越来越强。PLC 可以进行 PID 运算，以适应比例、积分、微分的要求。可进行 FCS 运算，为一组 ASCII 码进行纵向异或校验计算 FCS 码。

此外，还有种种逻辑运算指令基本逻辑指令。如，ANDW 字与、ORDW 字或、XORW 字异或及 XNRW 字同或。

总之，数据运算与处理指令是很多的。

2.3.4 流程控制指令

PLC 程序流程是可以控制的，其使用的指令主要有跳转、循环、子程序、中断、步进等。

1. 跳转指令

OMRON 机用的为 JMP（004）及 JME（005）。这两条要配对使用。

JMP 指令执行前，要建立逻辑条件。JME 不要条件，只是表示跳转结束。要跳转的程序列于这两个指令之间。

当执行 JMP 时，若其逻辑条件为 ON，则不跳转（注意：它与汇编语言的跳转含义相反，与有的 PLC 厂家的类似指令及后面介绍的 CJP 指令也相反），照样执行 JMP 与 JME 间的指令，如同 JMP、JME 不存在一样；若为 OFF，则 JMP 与 JME 间的程序不执行，有关输出保持不变。

图 2-38 具体说明了跳转指令的功能。

JMP、JME 可嵌套使用，但其层次有时受限制。

JMP、JME 编号使用时，配对的两个，编号要一致，而且不能重复。

有的可编程的跳转类似计算机的汇编语言，指定跳转到哪个标号的语句去执行，情况复杂一些，使用时要小心，要避免出现死循环。

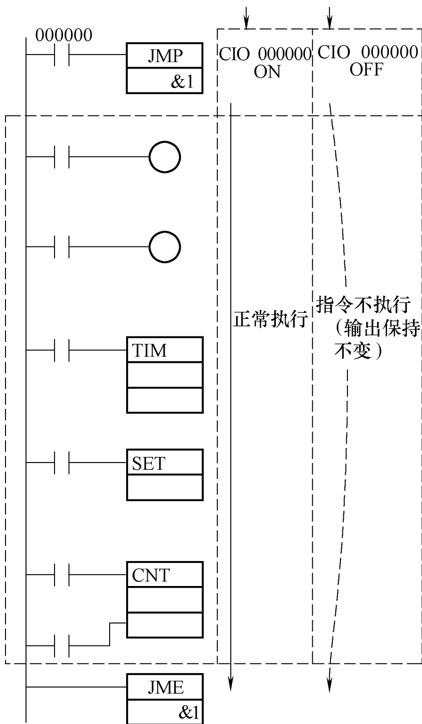


图 2-38 跳转指令使用示意图

CJ1 机还有其它几种跳转指令，CJP、CJPN、JME 等。其作用与 JMP 大同小异，具体如图 2-39、图 2-40 所示。

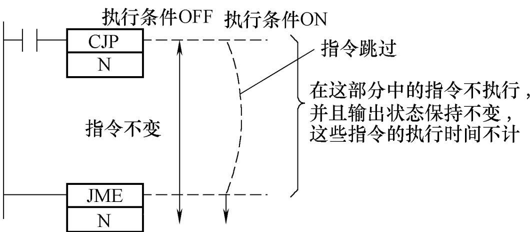


图 2-39 CJP 跳转指令使用示意图

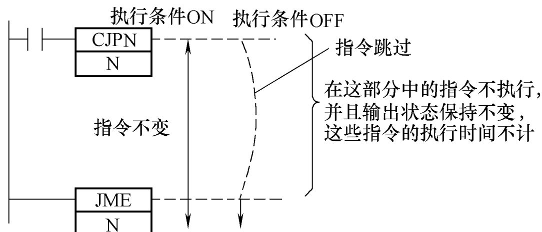


图 2-40 CJPN 跳转指令使用示意图

在此，附带对互锁 IL（002）和互锁清除 ILC（003）指令也作简要介绍。这两个指令在形式上，与跳转指令类似，也是要配对使用。但功能不同，它不改变程序流程，只是像电路的“总开关”一样，影响 IL 与 ILC 间的程序执行，如图 2-41 所示。

图 2-41 中“正常执行”意指，“总开关”合上，不影响 IL 与 ILC 间的程序执行。“输出互锁”意指，“总开关”断开，IL 与 ILC 间的程序执行条件全为 OFF，即其间的输出全被互锁住。

一般讲，IL（002）和 ILC（003）要配对使用，但有时也有例外。如图 2-42 所示，这里有多于一个 IL（002）和单个 ILC（003）配对。当执行程序检查时，会显示有错，但程序仍能正确执行。

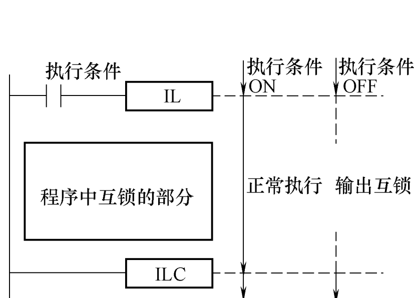


图 2-41　互锁和互锁清除指令示意图

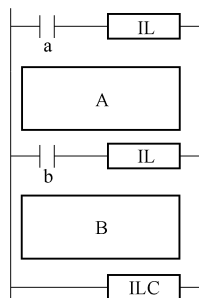
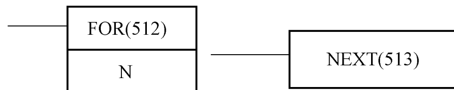


图 2-42　互锁和互锁清除指令例外使用

2. 循环指令

它由 FOR（512）和 NEXT（513）两条指令组成。两条指令要配对使用。其功能是，使这两条指令间的指令，按指定的次数，重复执行。

这两条指令的梯形图格式为



这里，N 为指定的重复次数。

不过已进入循环后，也还可用 BREAK（514）指令，使其退出循环。图 2-43 示出，如 a OFF，BREAK（514）指令不执行，则 FOR 与 NEXT 间的指令执行 3 次（这里，$N=3$）。但，如在执行中 a ON，则 BREAK（514）指令执行，程序将退出循环执行。

FOR-NEXT 循环可嵌套，可多至 15 层，在图 2-44 所示的例子中，程序段 A、B 和 C 如下执行：A→B→B→C，A→B→B→C 和 A→B→B。

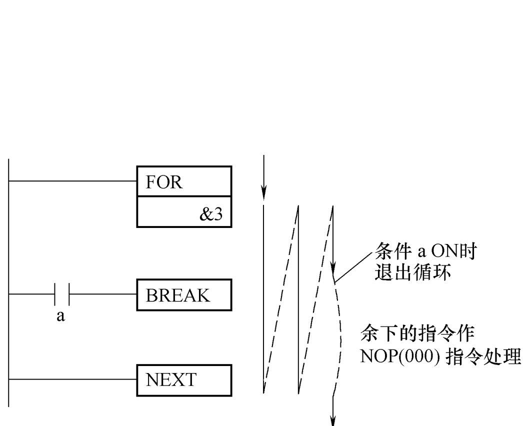


图 2-43　用 BREAK 指令退出循环

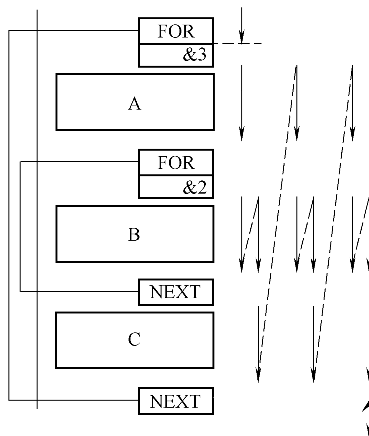


图 2-44　FOR-NEXT 嵌套使用

多层循环时，也可用 BREAK（514）从一个个 FOR-NEXT 循环中退出。若要从嵌套循环中退出，则需要多个（嵌套层数）BREAK（514）指令。图 2-45 所示为在多层循环中，使用 BREAK 指令退出循环的情况。

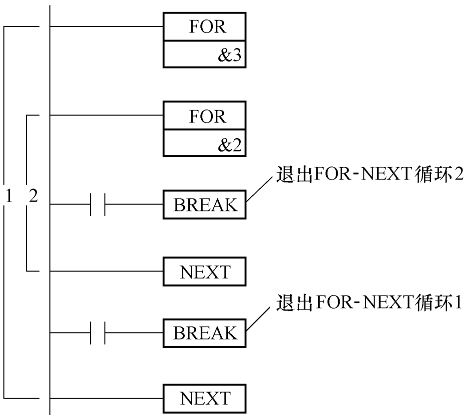


图 2-45 FOR-NEXT 多层循环

3. 子程序

在程序中，常有一些要重复使用的一组组指令，用以实现某些特定的功能。若把这一组组指令编成子程序，则可大大简化编程。

子程序的指令有 3 条：

SBN（92）N——子程序的入口；

RET（93）——子程序结束；

SBS（91）N——子程序调用。

N 为子程序编号，编号数与可编的子程序对应。

子程序入口及结束指令无须逻辑条件，而调子程序的指令要有逻辑条件。当逻辑条件 ON 时，执行 SBS 指令，则停止 SBS 之后的指令执行，转而去执行从子程序入口开始的程序。直至 RET 指令，则又返回主程序，执行 SBS 之后的下一个指令。若条件为 OFF，子程序不调用，如同 SBS 指令不存在一样，程序执行 SBS 之后的次一个指令。

使用子程序的情况如图 2-46 所示。

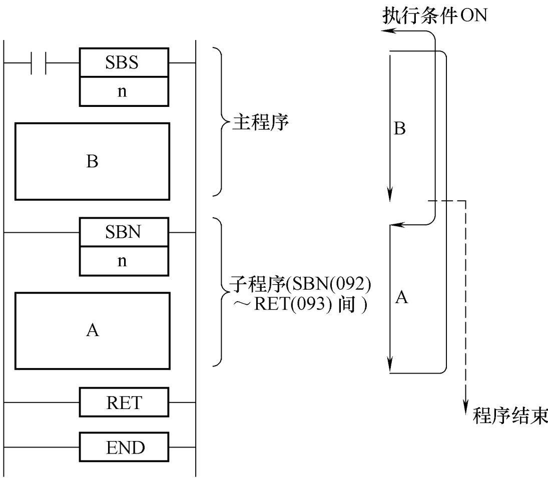


图 2-46 子程序使用示意图

SBS 指令可多次使用，亦即子程序可多次被调用。子程序被调用的次数越多，其使用的效率也越高，越说明子程序存在的必要。

子程序可以嵌套使用，即子程序还可调子程序。但其层数是受限制的，而且也不能调用自身，即不能递归。

图 2-47 所示为子程序嵌套使用的情况。这里，000000 ON 时，调子程序 1，子程序 1 执行时，如 000001 ON 则又调子程序 2。

除了子程序指令，OMRON 还有宏（MCRO（099））指令。实质上它也是子程序。不同的在于 MCRO（099）可以使用 S 和 D 中的参数改变子程序中位和字地址，而不改变子程序的结构。也就是说，它是带参数的调子程序。

提示：所有的子程序都要安排在主程序的后面，在 END 指令之前。

CJ1 机为多任务编程。这里讲的子程序只是局部的。只能在本任务中调用。要想所有任务中调用，要用全局子程序。

全局子程序指 GSBN（751）和 GRET（752）之间的程序段。其调用要用全局子程序调用指令 GSBS（750）。

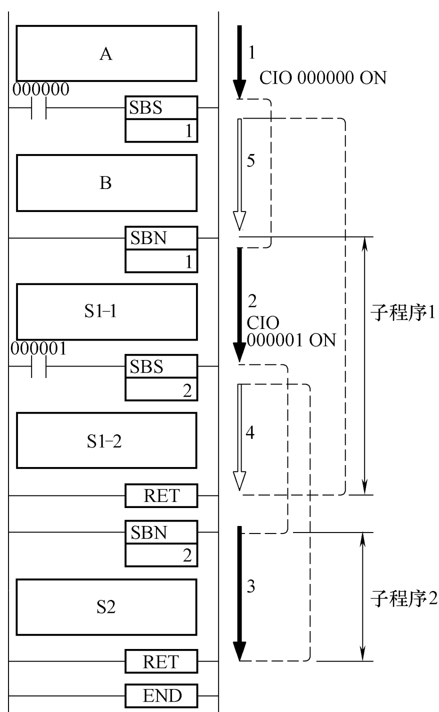


图 2-47　子程序嵌套使用

图 2-48 所示为全局子程序及其调用情况。这里循环或中断任务 1 与循环或中断任务 2，只要条件具备，均可调用全局子程序 n。

4. 中断

中断也是调子程序，但它不是靠 SBS 指令调，而是靠中断事件调。且调的子程序编号与所发生的事情对应。这些子程序有时还称为中断服务程序。

提示：CJ1 机是多任务编程的，它不调子程序，而是调中断任务，这点要注意。凡为多任务组织程序的 PLC，多也是如此。

PLC 中断事件可以来自外部，也可来自内部。前者称外中断，后者称内中断。

外部中断用输入点，如 CQM1 机，可指定 00000 ~ 00003 四个点为中断输入点；可编写四个中断服务程序与这 4 点从 OFF 到 ON 的事件对应。只要这四点之一从 OFF 转为 ON（产生中断事件），则调用一次对应的服务程序。

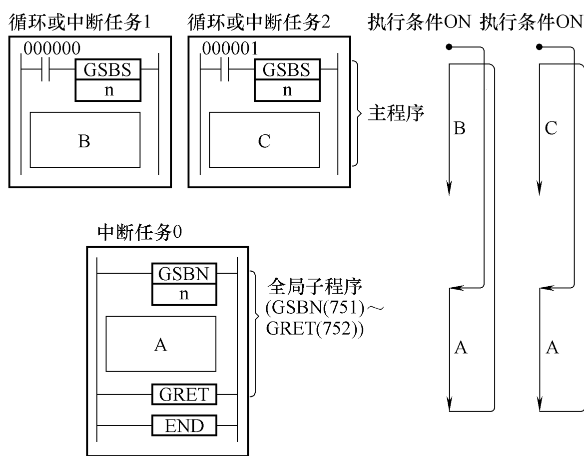


图 2-48　全局子程序调用

这种外中断，可缩短 PLC 对输入信号的响应时间。

内部中断的事件来自 PLC 内部，如定时中断。C200H 机，可通过 FUN（89）指令，设定时中断的周期。只要执行 FUN（89）一次，则 PLC 将定时地执行定时服务程序（子程序编号指定为 99）。CQM1 可通过 ST1M 指令设这个中断周期。其它 PLC 也有类似指令。

PLC 处理中断事件是有个过程的。当发生中断事件时，PLC 总是先记录发生的事件，并对其按优先级进行排队。优先级高的先执行，它执行完了，再执行优先级低的。所有中断任

务处理完了，再转回执行正常的循环程序。而优先级与中断的任务号是对应的。号越小，优先级越高。

要注意的是，已记录但未执行的中断，其后又发生相同的事件，PLC 对此将不理睬。所以，不是发生的所有中断事件都会处理的。

为了处理好中断，提高程序的控制可靠性与效率，PLC 还提供一些有关的中断处理指令。

在主程序中，某一部分不允许被中断，这时可用中断禁止指令；某一部分允许被中断，这时可用中断允许指令。在不使用这两个指令时，默认为中断允许（S7-200 默认为中断不允许）。

CJ1 机中断允许禁止的梯形图符号如下：



当程序运行在中断禁止区时，即使出现中断事件，正运行的程序也不中断。待进入允许区后才转去处理中断。

中断允许的梯形图符号如下：



使用这两个指令时中断允许、禁止的情况如图 2-49 所示。

中断可开通，也可被屏蔽。中断开通后，中断事件出现，才可去执行中断任务或子程序。中断被屏蔽时，中断事件即使出现，其相应的服务也执行。

屏蔽，开通有的靠指令，有的靠设定，使用时，要按说明书要求进行。CJ1 机用 MSKS 指令。其梯形图格式如下：

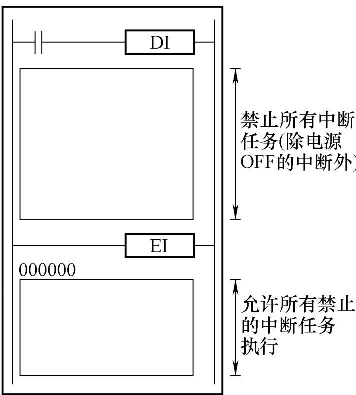
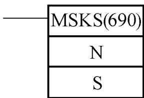


图 2-49 中断允许、禁止示意图



这里，N 为中断单元号；

S 为功能字。

当 N 为 0 或 1 时，S 的对应位的内容，决定中断单元 0 或 1 的对应点，是中断屏蔽（值 1 时），还是中断开通（值 0 时）。

如图 2-50 所示，当 000000 ON 时，执行一次本指令，则中断单元 0 的第 3、4、6、7、14、15 的输入点中断开通恢复，其它的中断屏蔽。

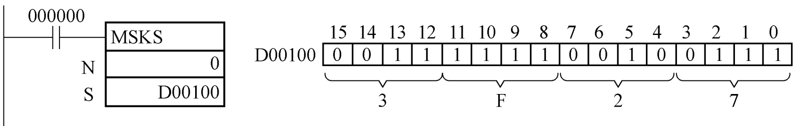


图 2-50 MSKS 指令使用例子 1

此外，还要可用 MSKS 指令，确定中断事件产生的时刻。这时 N 设为 2（对中断模块 0）或 3（对中断模块 1）。S 的对应位的内容，决定中断单元 0 或 1 的对应点，中断事件产生的时刻，是输入点 OFF 到 ON 时（值 0 时），还是 ON 到 OFF（值 0 时）发生事件。

如图 2-51 所示，当 000001 ON 时执行一次本指令，则中断单元 0 的第 1、8、10 点为从 ON 到 OFF 时，发生输入中断事件，而其它点为从 OFF 到 ON 时，发生输入中断事件。

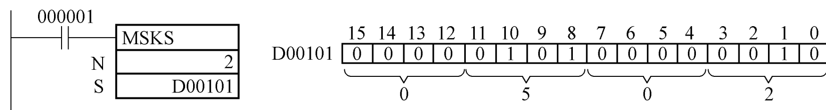


图 2-51　MSKS 指令使用例子 2

还可用本指令，开通定时使中断。这时 N 设为 4（用于定时中断 0）或 5（用于定时中断 1）。S 的值，再乘设定单位（可为 10ms、1ms，由编程软件设定）则为定时中断的间隔时间。如图 2-52 为 100ms。

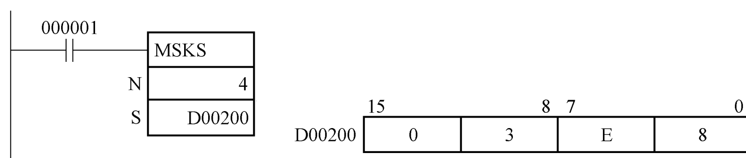
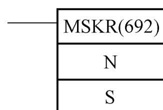


图 2-52　MSKS 指令使用例子 3

以上例子，说明执行 MSKS 指令是重要的。只有执行了 MSKS 指令后，才使中断生效。

此外，还有读中断屏蔽，MSKR 指令。其梯形图格式如下：



这里，N 为中断单元号；

S 为存储字。

执行本指令，将读出 N 指定的中断模块各屏蔽位的情况，并存于 S 中。当 N 为 0 或 1 时，S 的对应位的内容，决定中断单元 0 或 1 的对应点，是中断屏蔽（值 1 时），还是中断开通（值 0 时）。

如图 2-53 所示，当 000000 ON 时，执行一次本指令，则中断单元 0 的第 3、4、6、7、14、15 的输入点中断开通恢复，其它的中断屏蔽。

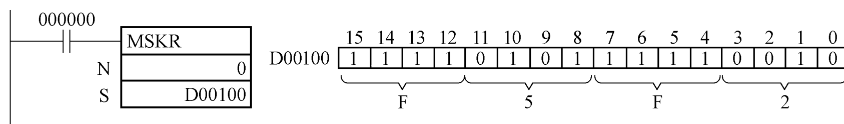
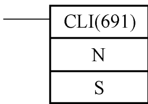


图 2-53　MSKR 指令使用例子

MSKR 指令中 N 的其它取值及其含义同指令 MSKS。与 MSKS 不同的只是，MSKS 是写（设定），而 MSKR 是读。

最后，还有 CLI 指令，用以取消已记录，但未执行的外中断任务。但不能清除一个正在执行的中断任务。它的梯形图格式如下：



这里，N 为中断单元号；

S 为设定字。

这时，N 取值及其含义与 MSKS 相同。S 值中某位为 1 时，相应的中断输入被清除。为 0，则其记录保留。

对定时中断，S 的值乘设定单位（可为 10ms、1ms，由编程软件设定）则为执行 MSKS 指令后第一个开始定时中断的时间间隔。如图 2-54 所示的说明。

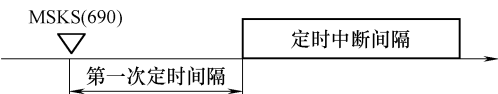


图 2-54　定时中断的时间间隔

如图 2-55 所示，当 000000 ON 时，本指令执行，可使执行 MSKS 后，要延长 50s（这里 1388 为十六进制数，折合十进制数为 5000，设定定时单位为 10ms），才再进入由 MSKS 设定的定时时间间隔定时中断。

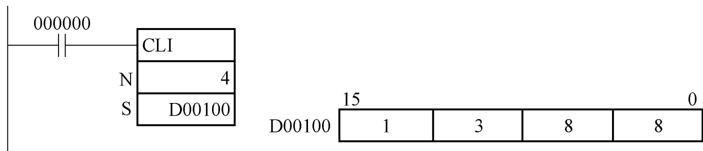


图 2-55　CLI 指令使用例子

如果在一个 I/O 中断任务正在执行时，接收到一个不同的中断输入，输入的中断号在内部被记录，直至当前任务和其它较高优先级别的任务执行完毕。而执行 CLI（691）可用于被记录的中断在其执行前将它们清除。

如执行图 2-55 中的 CLI（691）指令，它用的操作数为 D00101，其内容为 F2（十六进制），将中断输入单元 0 中除了输入 0、2 和 3 之外的所有记录的中断输入清除。如图 2-56 所示。

	F				2			
单元0的中断输入	7	6	5	4	3	2	1	0
中断清除/保留设置	1	1	1	1	0	0	1	0

1=清除记录的输入　　0=保留记录的输入

图 2-56　记录中断输入清除

在中断任务 3 执行完毕后，按优先级顺序执行已记录的中断。因为中断输入 0 记录了一个输入。则将在任务 3 执行完毕后执行 I/O 中断任务 0（中断任务 100）。因为 CLI（691）未保留中断输入 1，所以这一中断输入记录被清除，如图 2-57 所示。

在中断任务 3 执行完毕后，按优先级顺序执行已记录的中断。因为中断输入 0 记录了一个输入。所以将在任务 3 执行完毕后执行 I/O 中断任务 0（中断任务 100）。因为 CLI（691）未保留中断输入 1，所以这一中断输入记录被清除。

如中断输入 0 ~3 全变为 ON，且未执行 CLI（691），所有输入将被记录，并在中断任务 3 执行完毕后顺序执行之（中断任务按优先权从最低中断号到最高中断号顺序执行），如图 2-58 所示。

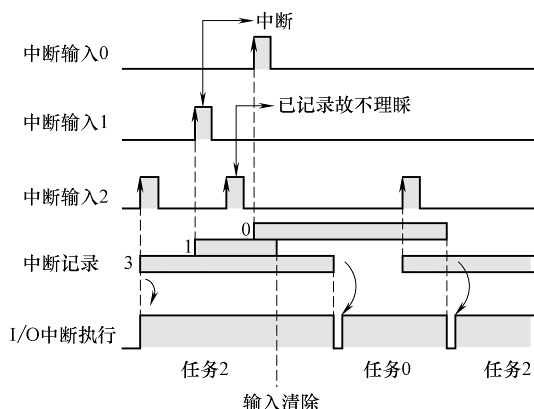


图 2-57　记录的中断输入清除示意图

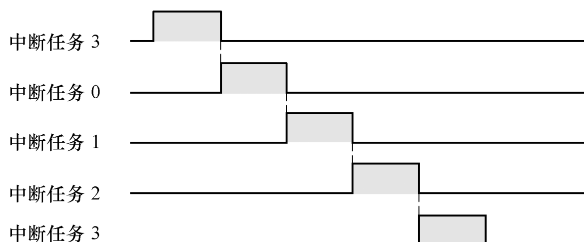


图 2-58　执行中断顺序

5. 步进指令

工作设备工作，或一些工艺过程往往分成步，按步进行。一个步完成了，再转入另一个步。可以有很多步。步，可以按顺序工作，也可依条件产生分支，改变顺序。需要时，还可多步并行工作……

为此，PLC 多设有步进指令。OMRON 机步进指令有两条：

STEP（008）B，步进程序入口指令。B 为标号，若没有标号 B，表示步指令结束。它只是表明步进程序，也称工序的入口，执行它无须逻辑条件。

SNXT（009）B，步进程序调用（激活）指令，调用标号为 B 的步进程序。执行它需逻辑条件。条件为真（ON），则执行调用，假（OFF）不调用。

有的 PLC，如日本三菱，B 为专用的继电器 S（状态继电器），OMRON 公司除输入继电器外，可用任何一种继电器。

步进指令的要点是：

1）只有已激活的步的程序才被扫描，才被执行；

2）在已激活步中，如激活了后续步，则自然处于非激活状态；

3）在程序中，可任意把某个步激活；

4）当某步激活后，原来激活它的条件变化，不再对其产生影响。因此，最好用脉冲信号建立逻辑条件。

步进程序可以顺序地被调用，直至最后一步。也可以分支调用，依条件按不同的分支进行。也可以平行调用，条件具备时，可同时调两个步程序，然后再依各步的情况再一步步推进，直到退出步进程序，返回主程序。

图 2-59 所示为步进指令的使用例子。

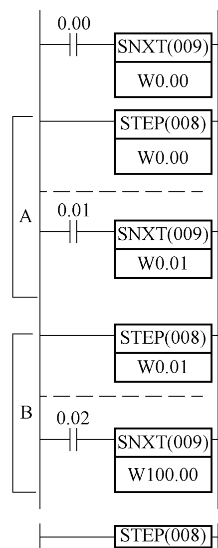


图 2-59　步进指令及其使用例子

从图知，它用 W0.00 等（内部工作区）作为步标识。当 0.00 ON，步 W0.00 被激活，A 段程序（在虚线处，程序未画出，下同）被扫描、被执行。这时，如 0.01 ON，则 W0.01 步激活，并退出 W0.00 步。同时，B 段程序被扫描、被执行。这时，如 0.2 ON，则 W100.00 步激活，并退出 W0.01 步。但，如程序中无此步，则意味

着退出步进程序。

提示：OMRON PLC 所有步进指令应放在其它主程序之后，子程序之前。如果主程序放在它之后，那这部分主程序将不执行；如把它放在子程序之后，那它也将不执行。

提示：OMRON 调新步时，旧的步先 OFF，然后，新的步才 ON。即新的与旧的步从不同时工作。而三菱 PLC、西门子 PLC，则是新的步先 ON 后，旧的步才 OFF，即新、旧的步将同时 ON 一个扫描周期。

2.3.5 功能块

功能块是 OMRON 公司新型 PLC 新增的程序模块。由一组指令及相应的操作数组成，可实现某种特定的功能。分系统功能块与用户功能块。前者由 OMRON 公司提供，用户可以调用，改变模块的属性为“显示功能块内部”后，还可看到原代码，但不可复制。后者由用户用梯形图或 ST 语言编写。

1. 系统功能块

系统功能块，在安装编程软件后，会自动加载到 OMRON 软件目录下的“Lib \ FBL \ omronlib”子文件夹中。而该文件夹下还有“PLC”、“Inverter”、“PositionController”、“TemperatureController”等若干子文件夹。这些子文件夹还含有多个子文件夹。如“PLC”文件夹下，就有“ENT”、“CLK”、“CPU”、“SCx”、“UNIT”、“CARD”等文件夹。在这些文件夹中，就有 cxf 文件。加载到工程中，每个文件就生成一个功能块，就可在工程程序中的调用。

调用功能块与调用定时器之类指令一样，也要指定实例名。同时，还要对功能块的输入赋值，对功能块的输出指定目标地址。

例如“PLC \ CPU”文件夹中的 CPU005_TOF_BCD.cxf 文件，加载后的系统功能块 CPU005_TOF_BCD，其功能是实现输入从 ON 到 OFF 时输出 OFF 的延时。实际是 OFF 延时定时器。OMRON PLC 原来只有 ON 延时定时器，有了此功能块也就有了 OFF 延时定时器了。图 2-60 所示为此功能块一个调用实例。

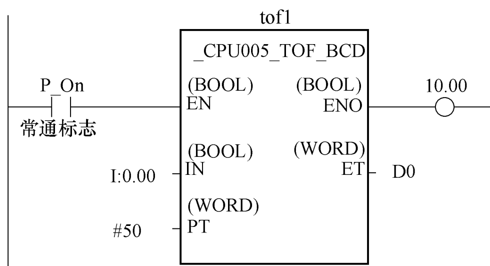


图 2-60 OFF 延时功能块一个调用实例

在该图程序中，“P_On”为常 ON 触点，以功能块使能“EN”ON，表示此功能块一直在调用。“tof1”为此功能块的实例名。功能块输入“IN”由“I:0.00”赋值，“I:0.00”ON，则“IN”ON，“I:0.00”OFF，则“IN”OFF。功能块输入“PT”，即定时设定值，为 BCD 码，单位为 100ms。在此赋值为常数 50，即设定延时 5s。功能块输出“ENO”指定的目标地址为“10.00”。功能块延时 OFF 就是由它实现。功能块输出“ET”，即延时现值，指定的目标地址为“D0”，也为 BCD 码，单位也为 100ms。

由此功能块的功能可知，执行此程序：“I:0.00”ON，“10.00”即时 ON；“I:0.00”OFF，“10.00”延时 5s 后 OFF。延时过程值将显示在“D0”中。

其它系统功能块各有各的功能，在此不再赘述。

可知，系统功能块实际上也是 PLC 指令，而且是功能更强的的指令。新增功能块可方

便地扩展 PLC 的指令系统，增加 PLC 的功能，使系统升级。而且，系统功能块是按需加载。不用的不加载，不占 PLC 内存。

提示：西门子 PLC 系统功能块多为不可视的。国产 PLC，LM 及 LK 机的系统功能块多是可视，又可复制。OMRON 系统功能块可视，但不能复制。

2. 用户功能块

用户功能块是传统子程序的扩展与增强。与子程序不同的是，它可以带参数（即，输入变量）调用（当然，也可不带），可以产生一个或多个外部输出（当然，也可不产生外部输出）。

如果不带参数调用，也不产生外部输出。那么，在功能上与子程序没有区别。

如果功能块只有一个外部输出，并能对若干输入变量按某个特定规律转换成这个输出，有的称为 PLC 函数，可按标准函数，如 SIN 函数，那样，赋值方法调用。但 OMRON PLC 没有这个函数概念，一律都称功能块，并都按功能块调用。

与子程序可调用另一个子程序，可以嵌套一样，功能也可调其它功能块。只是被调功能块必须同在一个工程中。

但是，在程序组织上功能块与子程序有很大区别。子程序与调用它的主（或另一子）程序总是组织在统一程序中，不能单独存储。而功能块可以扩展名为 cxf 的文件单独存储。也可单独加载，被别的工程引用。

可知，由于用户也可以开发自己的功能块，等于自己也可扩展所使用的 PLC 的指令系统。因而用好它，就等于 PLC 拥有指令的“条数”将不受限制。

要提及的是，子程序是早期 PLC 就已有的程序组织，而功能块是近期 PLC 才有的。所以，编写它要使用新版本的编程软件。同时还要针对新型的 PLC。此外，OMRON 功能块可以用梯形图语言编写，也可用 ST 语言编写。不过，一旦程序使用了功能块。则整个程序将无法使用助记符表达。

2.3.6　ST 语言简介

编写功能块可使用 ST 语言。ST 语言只有语句。语句类似梯形图程序的梯级。是一组含义完整的输入输出间变换的指令。OMRON ST 语言语句有：赋值语句、条件语句、选择语句、循环语句及其它语句。

OMRON 功能块必须先定义变量。有外部变量（系统提供的标志位）、输入变量（使用参数）、输出变量（返回值）、内部变量（在功能块中使用）及输入输出变量（兼有输入输出功能）。定义时，必须声明上述变量类型。如使用 ST 语言，可声明的类型见表 2-19。

表 2-19　ST 语言数据类型

数据类型	含义	位数	数 值 范 围
BOOL	位数据	1	0(FALSE),1(TRUE)
INT	整型数	16	-32768 ~ +32767
DINT	双字整型数	32	-2147483648 ~ +2147483647
LINT	长整型数	64	-9223372036854775808 ~ +9223372036854775807

（续）

数据类型	含义	位数	数 值 范 围
UINT	无符号整型数	16	&0 ~ 65535
UDINT	无符号双字整型数	32	&0 ~ 4294967295
ULINT	无符号长整型数	64	&0 ~ 18446744073709551615
REAL	实数	32	$-3.402823\times10^{38}\sim1.175494\times10^{-38}$,0, $+1.175494\times10^{-38}\sim+3.402823\times10^{38}$
LREAL	长实数	64	$-1.79769313486232\times10^{308}\sim-2.22507385850720\times10^{-308}$,0, $2.22507385850720\times10^{-308}\sim1.79769313486232\times10^{308}$
WORD	字	16	#0000 ~ FFFF 或 &0 ~ 65535
DWORD	双字	32	#00000000 ~ FFFFFFFF 或 &0 ~ 4294967295
LWORD	长字	64	#0000000000000000 ~ FFFFFFFFFFFFFFFF 或 &0 ~ 18446744073709551615

内部变量也可以与 PLC 的实际地址关联，成为 AT 变量，也称直接变量，是 PLC 软器件的实际地址。此外，OMRON ST 语言定义的变量还可为数组，但只能是一维数组。数组成员可从 1 ~ 32000。

OMRON ST 语言使用常数的表示与梯形图语言的表示也不一样。具体见表 2-20。

表 2-20　ST 语言常数

	表达方法	实例（对值 12）
十进制数	直接写入数	12
十六进制数	16#跟以数值	16#C
八进制数	8#跟以数值	8#14
二进制数	2#跟以数值	2#1100

再强调的是，OMRON 的 ST 语言不能在主程序中使用，只能在编写功能块时使用。但国产 PLC，LM、LK 机可以在主程序中使用。

以下对 OMRON ST 语言常用的语句分别进行介绍：

1. 赋值语句

其格式为

变量 A：= 表达式；（ * 这是注解 * ）

它有被赋值变量（变量 A）、赋值符号（：=）、表达式、结束分号（；）及注解组成。注解不是必要的，而其它则都不可缺少。其含义是，进行表达式运算，运算结果赋值给被赋值变量。而表达式则是由变量、运算符及括号组成。这也就是以前介绍的起、保、停逻辑就用有赋值语句。ST 语言使用的运算符见表 2-21。

表 2-21　ST 语言使用的运算符

运 算 名 称	符　　号	数 据 类 型	优 先 级
括号			1
函数调用			2
指数	* *	REAL,LREAL	3
非运算	NOT	BOOL,WORD,DWORD,LWORD	4

（续）

运算名称	符　　号	数据类型	优先级
乘	*	INT,DINT,UINT,UDINT,ULINT,REAL,LREAL	5
除	/	INT,DINT,LINT,UINT,UDINT,ULINT,REAL,LREAL	5
加	+	INT,DINT,LINT,UINT,UDINT,ULINT,REAL,LREAL	6
减	-	INT,DINT,LINT,UINT,UDINT,ULINT,REAL,LREAL	6
比较	< , > , < = , > =	BOOL, INT, DINT, LINT, UINT, UDINT, ULINT, WORD, DWORD, LWORD, REAL, LREAL	7
相等	=	BOOL, INT, DINT, LINT, UINT, UDINT, ULINT, WORD, DWORD, LWORD, REAL, LREAL	8
不等	< >	BOOL, INT, DINT, LINT, UINT, UDINT, ULINT, WORD, DWORD, LWORD, REAL, LREAL	8
与	&	BOOL, WORD, DWORD, LWORD	9
与	AND	BOOL, WORD, DWORD, LWORD	9
异或	XOR	BOOL, WORD, DWORD, LWORD	10
或	OR	BOOL, WORD, DWORD, LWORD	11

此外，系统还提供有初等数学函数，可在表达式中使用。这些函数见表 2-22。

表 2-22　ST 语言使用的初等数学函数

函　　数	参数数据类型	返回值数据型	内　　容	实　　例
ABS(参数)	INT, DINT, LINT, UINT, UDINT,ULINT,REAL, LREAL	INT, DINT, LINT, UINT, UDINT,ULINT,REAL, LREAL	求绝对值	a: = ABS(b)
SQRT(参数)	REAL,LREAL	REAL,LREAL	求根号	a: = SQRT(b)
LN(参数)	REAL,LREAL	REAL,LREAL	求自然对数	a: = LN(b)
LOG(参数)	REAL,LREAL	REAL,LREAL	求对数(10 为底)	a: = LOG(b)
EXP(参数)	REAL,LREAL	REAL,LREAL	求指数(e 为底)	a: = EXP(b)
SIN(参数)	REAL,LREAL	REAL,LREAL	正弦函数	a: = SIN(b)
COS(参数)	REAL,LREAL	REAL,LREAL	余弦函数	a: = COS(b)
TAN(参数)	REAL,LREAL	REAL,LREAL	正切函数	a: = TAN(b)
ASIN(参数)	REAL,LREAL	REAL,LREAL	反正弦函数	a: = ASIN(b)
ACOS(参数)	REAL,LREAL	REAL,LREAL	反余弦函数	a: = ACOS(b)
ATAN(参数)	REAL,LREAL	REAL,LREAL	反正切函数	a: = ATAN(b)
EXPT(底,指数)	底:REAL,LREAL 指数: INT, DINT, LINT, UINT,UDINT,ULINT	REAL,LREAL	任意底指数	a: = EXPT(b,c)

还有数据类型转换函数。如 REAL_TO_INT（A），可把实数 A 转换为一个字长的整形数。但数据类型转换是有限制的。表 2-23 示出了这个限制。

表 2-23　数据类型转换限制

源	目　　标											
	BOOL	INT	DINT	LINT	UINT	UDINT	ULINT	WORD	DWORD	LWORD	REAL	LREAL
BOOL	No	No	No	No	No	No	No	No	No	No	No	No
INT	No	No	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
DINT	No	YES	No	YES	YES	YES	YES	YES	YES	YES	YES	YES
LINT	No	YES	YES	No	YES	YES	YES	YES	YES	YES	YES	YES
UINT	No	YES	YES	YES	No	YES	YES	YES	YES	YES	YES	YES
UDINT	No	YES	YES	YES	YES	No	YES	YES	YES	YES	YES	YES
ULINT	No	YES	YES	YES	YES	YES	No	YES	YES	YES	YES	YES
WORD	No	YES	YES	YES	YES	YES	YES	No	YES	YES	No	No
DWORD	No	YES	YES	YES	YES	YES	YES	YES	No	YES	No	No
LWORD	No	YES	YES	YES	YES	YES	YES	YES	YES	No	No	No
REAL	No	YES	YES	YES	YES	YES	YES	No	No	No	No	YES
LREAL	No	YES	YES	YES	YES	YES	YES	No	No	No	YES	No

2. 条件语句

ST 语言有“假如、那么”语句。可用于逻辑处理。有 3 种格式：

（1）简单格式

```
IF <逻辑表达式> THEN
  <语句 1>
END_IF;
```

<逻辑表达式>为真，执行<语句 1>；否则执行“END_ IF”后续语句。

（2）分支格式

```
IF <逻辑表达式> THEN
  <语句 1>
ELSE
  <语句 2>
END_IF;
```

<逻辑表达式>为真，执行<语句 1>；否则执行<语句 2>。然后执行“END_IF”后续语句。使用它，可在<语句 1>与<语句 2>间选择。

（3）多重格式

```
IF <逻辑表达式 1> THEN <语句 1>;
ELSIF <逻辑表达式 2> THEN <语句 2>;
ELSIF <逻辑表达式 3> THEN <语句 3>;
...
ELSIF <逻辑表达式 n> THEN <语句 n>;
ELSE <语句 m>;
END_IF;
```

<逻辑表达式 1> 为真，执行<语句 1>。否则，<逻辑表达式 2>为真，执行<语句 2>。再否则，<逻辑表达式 3>为真，执行<语句 3>…直到<逻辑表达式 n－1>均为假，而<逻辑表达式 n>为真，执行<语句 n>。如上述逻辑表达式均为假，则语句执行<语句 m>。

图 2-61 所示的为 3 种格式条件语句的处理流程。

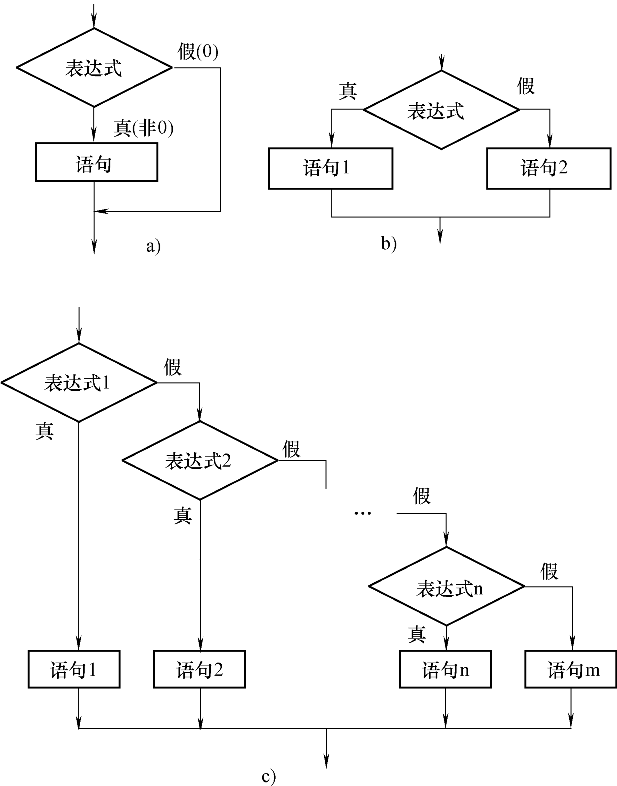


图 2-61 三种格式条件语句处理流程

（4）使用实例

1）例 1

```
IF A>0 THEN
    X:=10;
    ELSE
    X:=0;
```

```
END_IF;
```

上述语句含义为，如果变量 A >0，那么变量 X 赋值 10；否则赋值 0。

2）例 2

```
IF A >0 AND B >1 THEN
   X：=10；
   Y：=20；
ELSE
   X：=0；
   Y：=0；
END_IF;
```

上述语句含义为，如果变量 A >0 与变量 B >0 同时成立，那么变量 X 赋值 10，变量 Y 赋值 20；否则变量 X、Y 均赋值 0。

3）例 3

```
IF A THEN（＊这里 A 代表 A 为真＊）
   X：=10；
ELSE
   X：=0；
END_IF;
```

上述语句含义为：如果布尔变量为真，那么变量 X 赋值 10；否则赋值 0。

4）例 4

```
IF A >0 THEN X：=10；
ELSIF B =1 THEN X：=1；
ELSIF B =2 THEN X：=2；
ELSE X：=0；
END_IF;
```

上述语句含义为，如果变量 A >0，那么变量 X 赋值 10。否则，如果变量 B =1，那么 X 赋值 1；如果 B =2，那么 X 赋值 2。如果以上都不是，再变量 X 赋值 0。

3. Case（选择）**语句**

（1）格式

```
CASE 变量　OF
   值为 1:表达式 1；
   值为 2:表达式 2；
   值为 3:表达式 3；
ELSE 表达式 m；
END_CASE；
```

上述语句的含义为，当整形变量值为 1，执行语句 1。当整形变量值为 2，执行语句 2……余类推。如果没有合适的值，则执行语句 m。

（2）实例

```
CASE A OF
  1:X：=1；
  2:X：=2；
```

```
  3:X:=3;
ELSE X:=0;
END_CASE;
```

上述语句含义为，当 A 值为 1，X 赋值 1。当 A 值为 2，X 赋值 2。当 A 值为 3，X 赋值 3。如果不是上述值，则 X 赋值 0。

```
CASE A OF
    1:X:=1;
    2,5:X:=2;
    6..10:X:=3;
    11,12,15..20:X:=4;
ELSE Y:=0;
END_CASE;
```

上述语句含义为，当 A 值为 1，X 赋值 1。当 A 值为 2 或 5，X 赋值 2。当 A 值为 6 ~ 10 各值，X 赋值 3。当 A 值为 11 或 12 或 15 ~ 20 各值，X 赋值 4。如果不是上述值，则 X 赋值 0。

提示：CASE 变量必须使用整形数。即类型 INT、DINT、LINT、UINT、UDINT 或 ULINT。

4. 循环语句

循环语句可使一些语句重复执行。有 FOR loop、WHILE loop 及 REPEAT loop，与计算机高级编程语言循环语句相当。

（1） FOR loop

1） 格式

```
FOR 循环变量: = 初始值  TO 结束值 BY 增量值
DO
    语句 1;
    语句 2;
END_FOR;
```

上述语句的含义为，循环变量赋初值，执行语句 1、语句 2。执行后，循环变量加增量值，判断是否超过结束值。否，重复上述过程。是，跳出循环，结束循环语句。

2） 实例

（a） 例 1

```
FOR i:=0 TO 100 BY 1 DO
   array[i]:=0;
END_FOR;
```

上述语句含义为，数组 array 从下标 1 ~ 100 各元素都赋值为 0。

（b） 例 2

```
FOR n:=0 TO 50 DO(*增量值为 1,可以不写*)
   SUM:=SUM+DATA[n];
END_FOR;
```

上述语句含义为，累加数组 DATA 各元素的值。

（c） 例 3

```
MAX：=0；
MIN：=1000；
FOR n：=1 TO 50 DO(＊增量值为 1，可以不写＊)
    IF DATA[n] >MAX THEN
        MAX：=DATA[n]；
    END IF；
    IF DATA[n] <MIN THEN
        MIN：=DATA[n]；
    END IF；
END_FOR；
```

执行上述语句可把数组中最大值的数存入 MAX 中。把数组中最小值的数存入 MIN 中。

提示：循环变量及增量必须使用整形数，即 INT、DINT、LINT、UINT、UDINT 或 ULINT。

（2） WHILE 语句

1） 格式

```
WHILE 布尔表达式
    其它语句；
END_WHILE；
```

布尔表达式结果为真，执行 WHILE 到 END_WHILE 之间的语句。否则，转去执行后续语句。

2） 实例

```
WHILE counter < >0 DO
    Var1： = Var1 ＊2；
    Counter： = Counter -1；
END_WHILE
```

如果 Counter 初值设为 5，初始 Var1 为 1，上述语句执行，“Var1” 的值将是 32。

（3） REPEAT 语句

REPEAT 循环不同于 WHILE 循环，而是先执行，后判断。所以，它至少执行一次。

1） 格式

```
REPEAT
    执行语句；
UNTIL 布尔表达式
END_REPEAT；
```

2） 实例

```
REPEAT
  Var1： = Var1 ＊2；
  Counter： = Counter -1；
UNTIL Counter =0
END_REPEAT；
```

如果 Counter 初值设为 5，初始 Var1 为 1，上述语句执行后，“Var1” 的值将是 32。

5. 其它语句

（1） EXIT 语句

与 IF 语句配合，可根据条件终止重复语句执行。如下例，

```
FOR（或 WHILE,或 REPEAT)表达式
...
IF 布尔表达式 THEN EXIT;
END_IF;
...
END_FOR（WHILE, REPEAT）;
```

如上述布尔表达式真，不管重复执行的条件如何，都终止重复语句执行。

（2） RETURN 语句

用以结束本功能块，返回调用它的主程序。

（3） 功能块调用语句

不仅主程序可调用功能块。功能块也可用功能块调用语句，调用另一个功能块。调用前，要先声明一个“被调用功能块”类型的内部变量（工程中存在被调用的功能块才有此数据类型）。其格式为

```
FB2（参数 1，参数 2，参数 3…）;（* 如被调功能块变量名为 FB2 *）
```

参数传递都必须用本功能块的输入、输出变量（有的厂家 PLC 不受此限制）。具体有两种方法调用：

1） 语句传递法

```
FB2(EN:=bdata1,FB2_IN1:=FB1_IN1,FB2_IN2:=FB1_IN2,
    FB2_OUT1=>FB1_OUT1,FB2_OUT2=>FB1_OUT2,ENO=>bdata2);
```

这里，EN 及输出 ENO 两参数用本功能块的内部定义的布尔变量。输入用赋值传递：被调功能块输入变量 1：= 本功能块输入变量 1，被调用功能块的输入变量 2：= 本功能块输入变量 2 输出用指向传递：被调功能块输出变量 1 = > 本功能块输入变量 1，被调用功能块输出变量 2 => 本功能块输入变量 2……

2） 直接表示法

```
FB2(FB1_IN1,FB1_IN2,FB1_OUT1,FB1_OUT2);
```

这里，EN 及 ENO 不指定。被调功能块输入，直接用本功能块输入变量。被调功能块输出，直接用本功能块的输出变量。

要强调是，这两种方法不能混用。而且参数顺序要与调用功能块设定一致。

从以上介绍可知，ST 语言实际上是 PASCAL 语言的简化。功能比梯形图语言强得多。往往可先用它编写功能块，然后用梯形图程序调用它，可使 PLC 既实施灵活的控制，又可完成复杂运算。

提示：如同其它语言，同样为 ST 语言，各 PLC 厂家的细节，多不完全一样。如国产 PLC，LM、LK 机，可定义二维数组。还可定义结构变量、枚举变量及指针。运算符也更多些。

以上只是对 OMRON PLC 的有关指令作些简要地、汇总介绍。只是希望读者能从总体上，或从功能上理解它。实际上，即使都是 OMRON PLC，型号不同，版本不同，指令系统也不完全相同。所以，指令的细节，请参阅有关说明书。而且，对它的变化与进展也应时时留意。

2.4　PLC 典型程序

关键词：“起、保、停”程序、状态转换程序、定时控制程序、动作控制程序、步进程序、转换程序、联锁、互锁程序

本节将介绍一些 PLC 典型程序实例。为以后学习 PLC 编程建立基础。

2.4.1　起、保、停程序

任何设备，总有使其工作（起动）与使其停止工作（停止）的问题。本书第 1 章介绍一个最简单的、用两个按钮的起、保、停程序。除此，常见的有以下一些：

1. 单按钮起、保、停程序一

图 2-62 所示为单按钮起、保、停梯形图程序，操作数为符号地址。

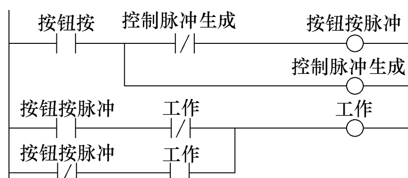


图 2-62　单按钮起、保、停程序一

从图知，当“按钮按”OFF 时，“按钮按脉冲”及“控制脉冲生成”均 OFF。而“按钮按”ON 时，“按钮按脉冲”、“控制脉冲生成”也 ON。但在下一个扫描周期时，因“控制脉冲生成”ON，而使“按钮按脉冲”将 OFF。即当“按钮按”ON 时，“按钮按脉冲”仅 ON 一个扫描周期。脉冲信号也因此得名。如所用得 PLC 有生成脉冲的指令，则也可直接用它生成脉冲。

当无脉冲信号，其“工作”的状态不会改变。因为这里的“工作”状态是“双稳”的，其为 ON 或 OFF 均成立。不妨看一下它的逻辑关系就清楚了。但一旦有脉冲信号作用，则其状态将改变。若开始为 OFF 将改变为 ON。反之，将改变为 OFF。也正因此，即可用这里的“起、保、停”控制这里的“工作”。

以下为图 2-62 梯形图程序转换成的助记符程序，但指令地址未标明。

LD　按钮按

OUT　TR0（因为这里梯形图存在分支，转换时需先用暂存器 TR0 存储“按钮按”信号状态）

ANDNOT 控制脉冲生成

OUT　按钮按脉冲

LD　TR0（这时，要取出暂存器 TR0 存储“按钮按”信号状态）

OUT　控制脉冲生成

LD　按钮按脉冲

ANDNOT 工作

LDNOT 按钮按脉冲

AND　工作

ORLD（因为以上梯形图有并联关系，故这里用逻辑块指令）

OUT　工作

END

用单按钮起、保、停，可节省 PLC 的输入点与按钮，还可简化操作面板的布置，PLC 程

序也不复杂，是较常用的起、保、停控制方法。

2. 单按钮起、保、停程序二

上述单按钮起、保、停梯形图程序，是起、是停，容易“糊涂”。其实，完全可使用按钮按下不同的时间，去区分是起、还是停。很多小仪器以至手机也多是这么处理的。图2-63所示即为这个程序。

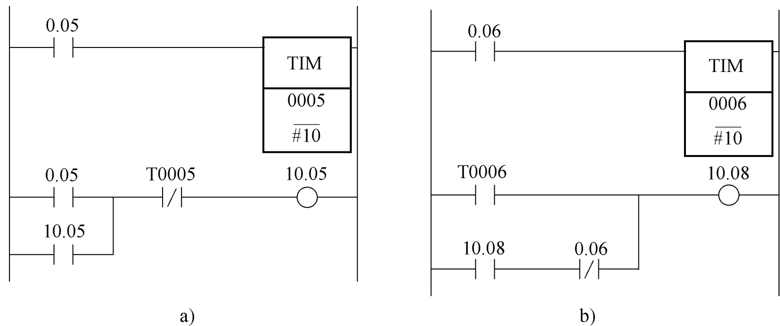


图 2-63　单按钮起、保、停程序二

a）短按起、长按停　b）长按起、短按停

图 2-63a 为短按起动、长按（超过 1s）停车程序。这里用了定时器 TIM 5，只有按钮按下超过 1s，它的常闭触点才分断，才能把已工作的输出 10. 05 停止工作。如果按下时间少于 1s，10. 05 将起动工作。

图 2-63b 为长按（超过 1s）起动、短按停车程序。这里用了定时器 TIM 6，只有按钮按下超过 1s，它的常开触点才接通，才能把未工作的输出 10. 08 起动工作。如果按下时间少于 1s，10. 08 将停止工作。

以下为图 2-63a 梯形图程序转换成的助记符程序，但指令地址未标明。

```
LD 0. 05
TIM 0005 #10
LD 0. 05
OR 10. 05
ANDNOT T0005
OUT 10. 05
```

以下为图 2-63b 梯形图程序转换成的助记符程序，但指令地址未标明。

```
LD 0. 06
TIM 0006 #10
LD T0006
LD 10. 08
ANDNOT 0. 06
ORLD
OUT 10. 08
```

3. 多点起、保、停程序

图 2-64 所示为“多点起、保、停”梯形图程序，操作数也是用符号地址。

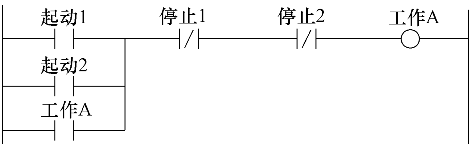


图 2-64　多点起、保、停程序

从图知，初始状态（所有输入、输出均未工作，下同）时，从梯形图逻辑知，“工作 A”为 OFF。这时，当“起动 1”或“起动 2”ON，而“停止 1”与“停止 2”又同为 OFF 时，则“工作 A”将 ON。一旦“工作 A”ON 后，即使“起动 1”或“起动 2”OFF，由于“工作 A”触点的“自保持”（它与“起动 1”、“起动 2”并联）作用，“工作 A”仍将保持 ON。

“工作 A”ON 后，如“起动 1”与“起动 2”同为 OFF，而“停止 1”或“停止 2”ON，则“工作 A”将 OFF。

由于“起动 1”、“停止 1”及“起动 2”、“停止 2”可布置在不同的位置，因而可用它在不同的地点对“工作 A”作起、保、停控制。

其实，图 2-62 的“按钮按”，也还可有多个并联的节点。并联后，其各触点信号 ON 时均可产生脉冲信号，都可实现对那里“工作”起、保、停的控制。那么做，图 2-62 即成为“多点单按钮起、保、停程序”了。

以下为图 2-64 梯形图程序转换成的助记符程序，但其指令编号地址未标明。

```
LD 起动 1
OR 起动 2
OR 工作 A
ANDNOT 停止 1
ANDNOT 停止 2
OUT 工作 A
END
```

4. “星-三角法”起动电动机的程序

图 2-65 所示为此程序，操作数也是用符号地址。

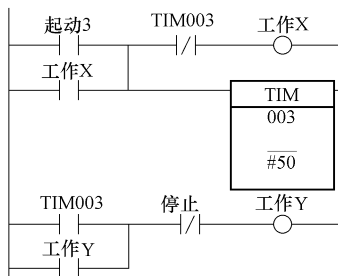


图 2-65　星-三角法起动电动机的程序

从图知，“起动 3”ON，“工作 X”将 ON，按星形接线起动。同时，定时器 TIM 003 开始计时（也可用速度继电器检测电动机转速，转速大到一定值，其触点 ON，代替这里的延时继电器触点 ON）。计时到，它的常闭触点使“工作 X”OFF，切断星形接线；它的常开触点使“工作 Y”ON，接通三角形接线。这样，即实现了“星-三角法”起动电动机。这样起动，可减少电动机起动电流，是很简便的电动机起动方法。

以下为图 2-65 梯形图程序转换成的助记符程序，但其指令编号地址未标明。

```
LD 起动 3
OR 工作 X
OUT TR0
ANDNOT TIM003
OUT 工作 X
LD TR0
TIM 003 #50
LD TIM003
OR 工作 Y
ANDNOT 停止 3
OUT 工作 Y
```

2.4.2 状态转换程序

工业控制对象往往处于多种状态下工作。为了适应各个工况的要求，在各状态间，常要进行转换。常见的有两种转换方法：

1. 直接转换

图 2-66 所示为“直接转换”梯形图程序，操作数也是用符号地址。

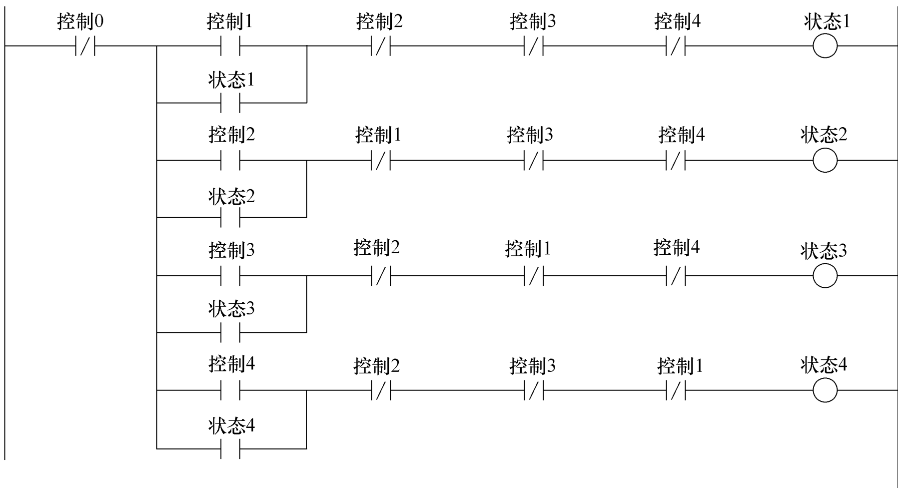


图 2-66 直接转换梯形图程序

从图知，任一控制信号（“控制 1、2、3、4”）ON，都将使原状态（“状态 1、2、3、4”）OFF，而进入新状态（ON）。可实现直接转换。

要退出所有状态，可使“控制 0”ON。

2. 先退出原状态，后转换

图 2-67 所示为“先退出原状态，后转换”梯形图程序，操作数也是用符号地址。

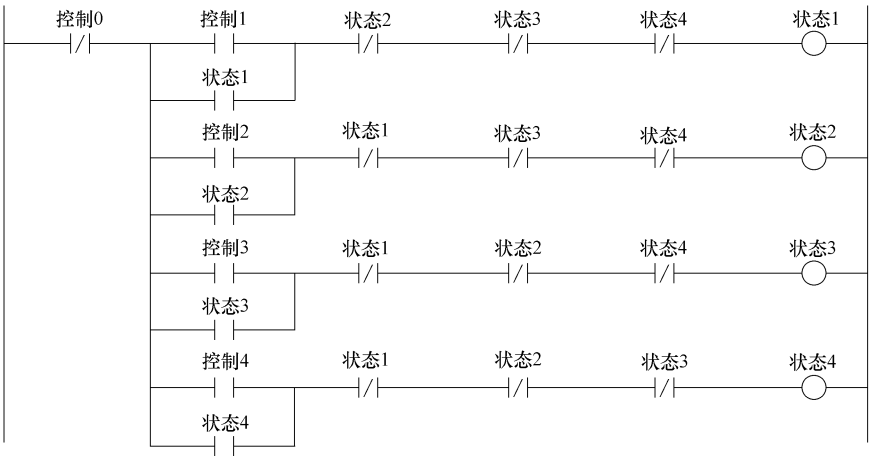


图 2-67 先退出后转换程序

从图知，如原来处于某一状态，任一控制信号（“控制 1、2、3、4”）都无法退出所处状态。只有先使“控制 0”ON，用它的常闭触点把原状态 OFF。才可能用新的控制信号，去使新的状态 ON。

为了确保状态转换安全，有的系统必须这么做。

2.4.3　定时控制程序

1. 定时器控制

图 2-68 所示为“定时器控制的定时控制”梯形图程序，操作数也是用符号地址。

从图知，电路起动后，“控制开始”ON。先是定时器 TIM 000 开始计时。TIM 000 计时到，其常开触点 ON，进而定时器 TIM 001 开始计时。TIM 001 计时到，其常开触点 ON，进而定时器 TIM 002 开始计时。TIM 002 计时到，其常开触点 ON，进而定时器 TIM 003 开始计时。等等。

随着这一系列定时器相继工作，将分出很多时间段。即可按需要产生相应的动作。如本例：“动作 0”出现时间段 1（从“控制开始”ON，直至 TIM000 定时到）。“动作 1”出现时间段 1（从 TIM 001 定时到开始，直至 TIM 001 定时到）。等等。

这里“动作”也可不这样对应，有的动作还可在多个时间段出现，这些，完全可依实际需要决定。这里的实质是，各个动作都是按时间的推进，逐步出现的。是由时间信号控制的。

2. 计数器控制

时间控制还可用计数器加时间脉冲信号实现。图 2-69 所示为“计数器控制的定时控制”梯形图程序，操作数也是用符号地址。

从图知，“开始”ON 后，“工作”被置位（执行 SET 指令）。可逆计数器 CNTR 020 每隔 0.1s 加 1。“工作”ON，将执行计数器 CNT 的内容与设定值（此处为即时数 599，BCD 码）比较。当 CNT 的内容小于设定值，则比较标志位“P_LT”常开触点 ON，进而使“时间段 1”ON。这将产生与“时间段 1”ON 对应的动作。当 CNT 的内容不小于设定值，则比较标志位“P_LT”常开触点 OFF，进而使“时间段 1”OFF。而“P_LT”常闭触点 ON，则使“时间段 2”ON。这将产生与“时间段 2”ON 对应的动作。

这里只有两个“时间段”。其实，可增加使用比较指令次数，作多个比较，以得到多个时间段。这样，就完全可依实际的时间段的不同，产生不同的动作。这里的实质也是，各个动作都是按时间的推进，逐步出现的。也是由时间信号控制的。

2.4.4　动作控制程序

动作控制是很常见的控制逻辑。只要用这里的“动作”去控制对应输出点，再用对应的输入点去控制这里的“动作完成”，就可实现如机床刀架运动那样部件的自动控制。

还要指出的是，动作控制也可插入定时控制。如用某“动作”去控制某输出点的同时，还去起动一个定时器，令其计时，再用该定时器计时到信号作为该动作完成信号，则这一动作就是定时控制。这种逻辑也是用得很多的。

1. 半自动工作

图 2-70 所示为半自动工作的“动作完成控制”梯形图程序，操作数也是用符号地址。

从图知，“起动”ON 后，“动作 0”ON，并自保持。这里把“动作 1”、“动作 2”、“动作 3”的常闭触点串入，目的是一旦进入工作，而又完成所有动作，则不允许“动作 0”再被起动。

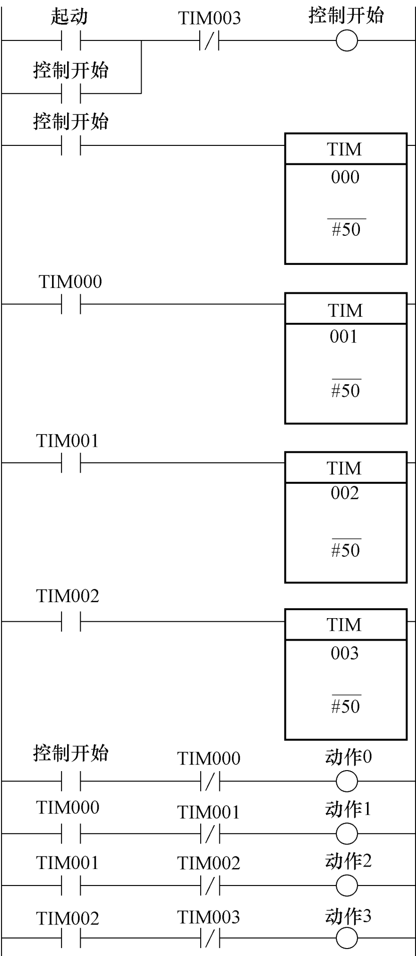


图 2-68　定时控制程序 1

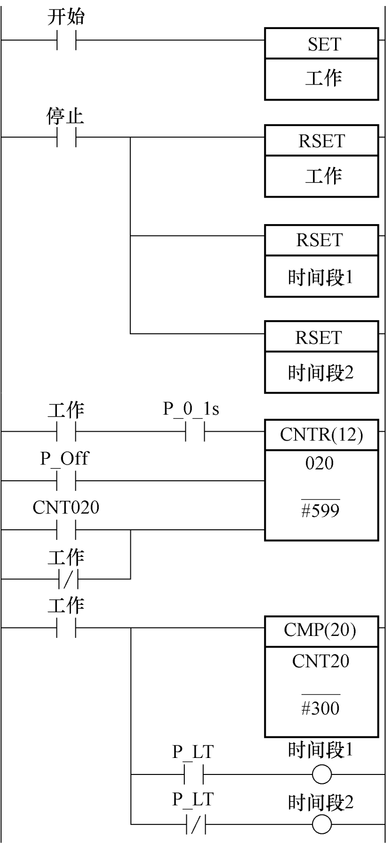


图 2-69　定时控制程序 2

"动作 0"ON 后，则执行与"动作 0"信号对应的动作，直到这个动作完成。当 PLC 检测到"动作 0 完成"信号，即"动作 0 完成"ON，则"动作 1"ON，并自保持。起动了与其相应的动作。另外，"动作 0"ON，还使"动作 0"OFF，有关它的动作停止。

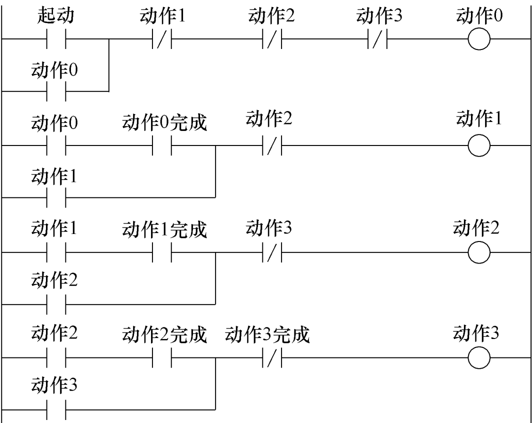


图 2-70　动作半自动控制程序

动作 1 完成，也将起动动作 2……直到动作 3 完成，则程序回到原状态。

可知，此程序实现的动作转换，是半自动控制。

2. 自动工作

图 2-71 所示为自动工作的"动作完成控制"梯形图程序，操作数也是用

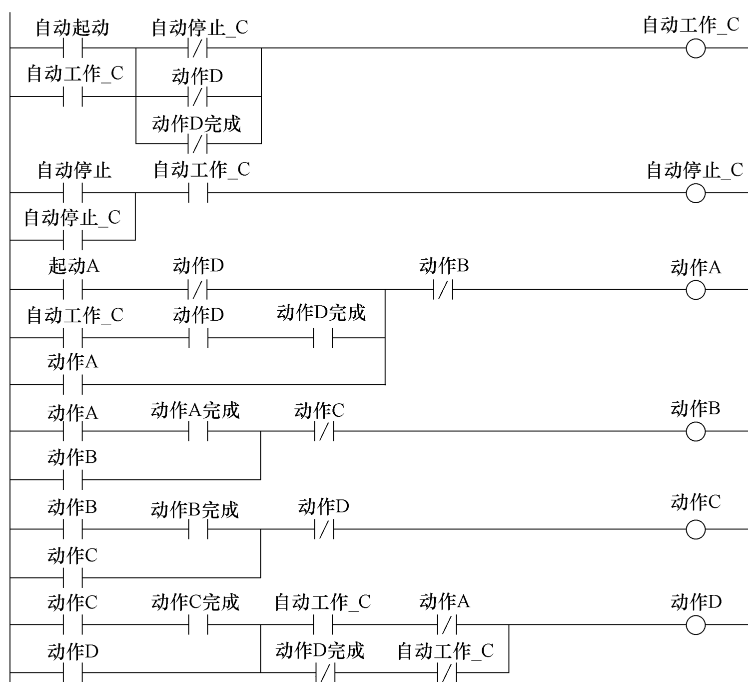


图 2-71　动作自动控制程序

符号地址。

从图知，这里多了“自动起动”、“自动停止”及有关信号。“自动起动”ON，将使“自动工作_C”ON，即起动了“自动工作逻辑”，并自保持。如“自动工作_C”ON，且“自动停止”ON，将使“自动停止_C”ON，并自保持。这将为停止自动工作的逻辑作准备。

当未起动“自动工作逻辑”时，此程序与“半自动”程序是完成相同的。只是这里用了“ABCD”，而“半自动”程序用的是“0123”。

当起动了“自动工作逻辑”时，“动作 D 完成”ON，将起动“动作 A”，再由“动作 A”的常闭触点去使“动作 D”OFF。起动“动作 A”，意味着新的循环开始。

这时，若要停止继续工作，可使“自动停止”ON。“自动停止”ON，将使“自动停止_C”ON，并自保持。若如此，则在“动作 D 完成”ON 时，将使“自动工作_C”OFF。这意味着，这时“动作 A”不能再起动。而“自动工作_C”OFF 也使“自动停止_C”失去自保持，程序将回到原状态。

可知，此程序进入“自动工作逻辑”后，动作是周而复始地自动执行着。而要退出“自动工作逻辑”，则应使“自动停止”ON。而且，要到所有动作完成后，即动作 D 完成后，才能完全退出“自动工作逻辑”，并停止所有动作。

2.4.5　步进程序

定时控制简单，而又能完成较复杂的控制。但它没有反馈，是开环控制。前一时间段动作完成与否，次一时间段的控制命令照样发出。工作可靠要求很高的场合不好用它。

动作控制是反馈控制，前一个动作未完成，次一个动作不会开始。较安全、可靠。但用

它去实现较复杂的控制比较难。

这里引入的步进程序，靠步的推进实现控制，而步的推进则用“步动作完成”信号激发。这种控制可用步控制输出，能实现复杂动作。同时，他又有“步动作完成”信号反馈，是闭环控制，较为可靠。

实现步进控制的方法很多，这里仅介绍两种较简单的方法。

1. 脉冲步进

图 2-72 所示为“脉冲步进”梯形图程序，操作数也是用符号地址。它是靠基本逻辑指令实现步进控制。

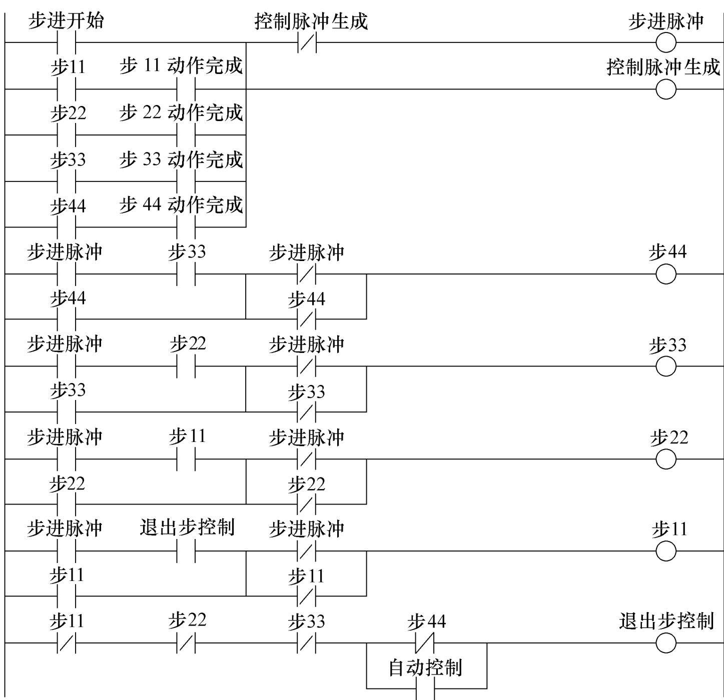


图 2-72　脉冲步进程序

从图知，此程序主要由脉冲生成及步进两个部分组成。

在脉冲生成部分可知，当“步进开始”或“步”及相应“步动作完成”信号出现，都将产生“步进脉冲”信号。

在步进部分可知，这里暂设了四个步：“步 11”、“步 22”、“步 33”、“步 44”。如需要可增加。“步 11”为初始步。当所有步未激活（均为 OFF），则“退出步控制”为 ON。

此时，如“步进开始”ON，则将有“步进脉冲”ON 一个扫描周期。这将使“步 11”激活，处于 ON 状态，进入工作。而其它步仍为 OFF。

当步 11 动作完成后，“步 11 动作完成”ON，将再有“步进脉冲”ON 一个扫描周期。这将使“步 22”激活，处于 ON 状态，进入工作。而使“步 11”OFF，其它步仍为 OFF。即步控制前进了一步。

这里每一步的控制逻辑与“单按钮起、保、停”逻辑形式不同，但本质是相同的。“各个步”也是“双稳”，无“步进脉冲”信号，它 ON 成立，OFF 也成立。一旦有脉冲信号，则转换成相反的状态。

随着步动作完成而产生的脉冲信号，步将逐一推进，直到最后一步。最后一步完成时，如“自动控制”ON，则又起动第一步；如“自动控制”OFF，则推出步进控制，程序回到原状态。

提示：这里步程序是按倒序排列的，控制“步 44”逻辑，反而排在“步 11”之前。可检查，如不是这么安排，按步推进的目的是达不到的。这也说明 PLC 指令的安排顺序是有讲究的。

2. 移位步进

图 2-73 所示为“移位步进”梯形图程序，操作数也是用符号地址。它是靠移位指令（SFT）实现步进控制。

从图知，此程序由四个梯级组成。

第一梯级，用以产生“移位脉冲”信号。

第二梯级，用以当步进完成后（这里设了四步，如需要，可增多），复位，把 0 传给“移位通道”。

第三梯级，用以产生“移位通道等零”信号。在“移位通道”字的内容为零时，“移位通道等零”为 1。

第四梯级，用以实现移位步进。这里的复位信号为“P_Off”（常 OFF），故只要“移位脉冲”从 0 转到 1，则把“移位通道等零”的状态（0，或 1）移入“移位通道”的第 0 位，而原“移位通道”的第 0 位状态，移给“移位通道”的第 1 位……依次移位，直到“移位通道”的第 15 位溢出。

显然，它与第三梯级配合将是，当“移位通道”为 0 时，“移位脉冲”从 0 转到 1，向“移位通道”移入 1；而当“移位通道”移入 1 后，移入 0；直到复位。

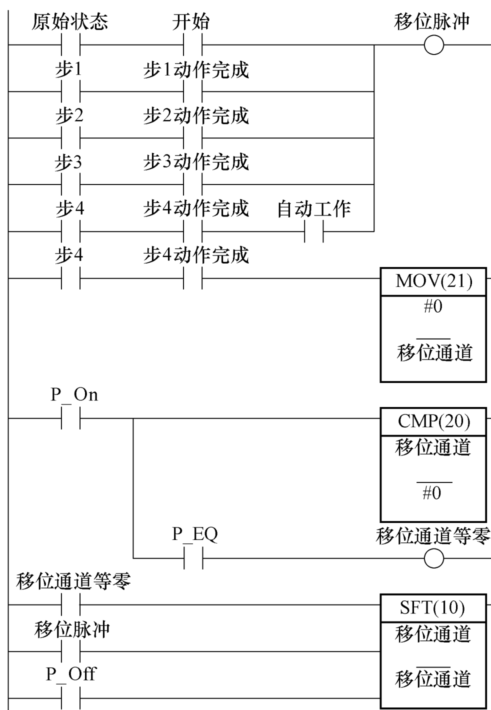


图 2-73　移位步进程序

可知，只要把“移位通道”0 位对应于步 1，1 位对应于步 2……则这里的移位过程，也就是步进过程。这里“移位步进”也因此得名。

这里还有“自动工作”控制。它 ON 时，将实现自动工作，即完成最后一步时，会产生“移位脉冲”，起动第一步。

2.4.6　转换程序

1. 输出转换程序

用以上介绍的“状态”、“时间段”、“动作”或“步”去控制输出点，以产生控制的实际的物理输出。有一对一转换（见图 2-74），一对多、多对一（见图 2-75）等转换。

这里，为了安全，用“IL”及“ILC”互锁指令，增加了急停控制。

2. 输入转换程序

用输入点去控制以上介绍的“控制”、“动作完成”，以发挥输入点对系统的实际控制作用。也有一对一转换，一对多、多对一等转换。与输出转换不同的只是把图 2-74 及图 2-75 的“状态”改为“输入”，“动作”改为“控制”或“动作完成”。

一经以上转换，即可按要求，用实际输入进行控制。

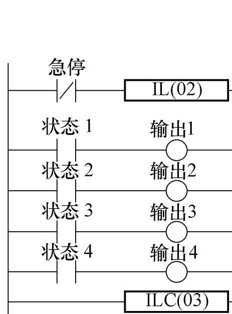


图 2-74 一对一转换程序

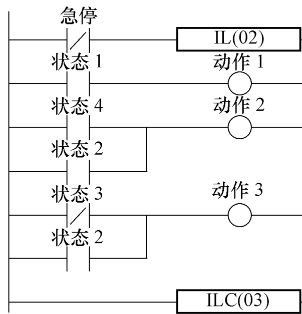


图 2-75 一对多、多对一转换程序

3. 转换参数选择程序

如以上介绍的“状态”、“时间段”、“动作”或“步”，若要选用工作参数，如温度、压力、时间等物理量的对应值，即可用此电路把它们与对应的量关联。这种选择电路也叫“配方”选择，以使控制系统能适合不同工艺参数的要求。

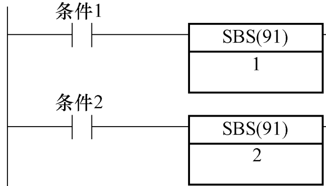


图 2-76 调用不同的子程序

4. 转换选择程序的程序

用此程序，把上述选择程序根据相应的条件再作选择，以实现多品种、多工艺、多参数的控制。实现这种选择的选择有两个办法：

用子程序（见图 2-76），调用不同的子程序作不同的选择；

用程序跳转，靠不同的程序跳转作不同的选择。

2.4.7 联锁、互锁程序

联锁、互锁是生产现场常见逻辑关系，用得是很多的。

1. 联锁

以甲的状态工作作为乙的工作前提条件，称甲对乙的联锁。实现的办法是用甲处工作状态作为乙工作的起动或工作条件。如图 2-77 所示，其逻辑关系可保证，只有甲工作（该图为动作 3）后，才可能产生动作 4（乙工作）。即乙被甲所联锁。

2. 互锁

互以对方的不工作作为自身工作的前提条件。实现的办法是，以对方的不工作作为自身的起动条件。如图 2-78 所示，动作 3 与动作 4 不能同时出现，只有一个动作停止后，另一个动作才能产生。电动机正反转控制就要使用互锁。

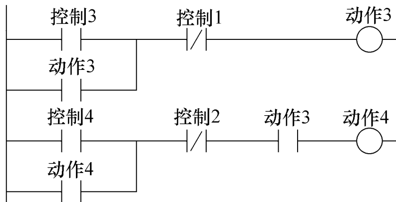


图 2-77 联锁程序

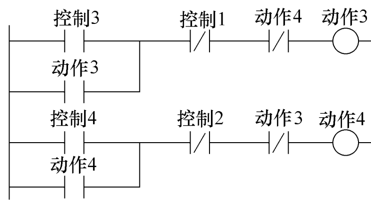


图 2-78 互锁程序

2.5　PLC 编程工具

关键词：简易编程器、图形编程器、编程方式、监控方式、运行方式、I/O 表登记、密码、解密操作

2.5.1　简易编程器

简易编程器是最常用的编程工具。

1. 结构

图 2-79 所示为 C200H-PR027 手持式编程器，通过电缆接于 PLC 上。适用于 OMRON 公司的多种型号的 PLC。此外，还有 CQM1H-PR001 及 CQM1-PR001 型号的简易编程器。结构类似，但它自带有电缆。

它的面板由三部分组成：

LCD 显示部分：有两行，每行可显示 16 个字符，相当于微型计算机的显示器，用以显示信息。

方式切换开关：用以控制 PLC 的工作状态，可使 PLC 处于编程、监控或运行三者之一状态。

键盘部分，如用原来的面板，如图 2-80 所示。如安装 CS1W-KS001-E 操作面板纸（日文显示），如图 2-80b 所示。

有近 40 个按键，原来的面板，可分为四类：

数字键：分布在操作键的左下方，白色键。有 0、1、…、9、10 个。其中 0、1、…、5，还可兼而对应输入 A、B、…、F。为后者时，应先键入上档（SHIFT）键（于操作键的左上角）。SHIFT 键的作用，如同计算机键盘上的 SHIFT 键。

指令键：分布在操作键的左上方，灰色键。用于输入 PLC 的指令，如 LD（装载）、OR（或）、AND（与）、NOT（非）、FUN（功能）等。

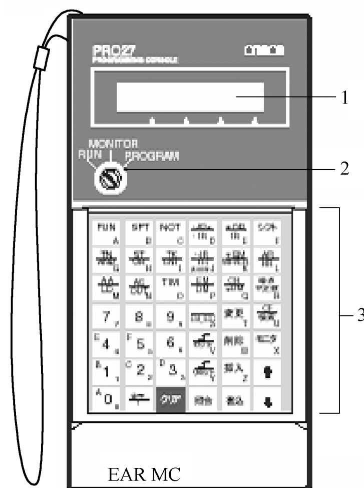


图 2-79　简易编程器
1—液晶显示屏　2—PLC 模式控制开关　3—键盘

数据标识键：分布在键盘部分的右上方，灰色键。用于标识内部器件的类型，如 DM、HR 等。

控制键：分布在右下方，黄色键。用以控制编程或监控操作，如 WRITE（写入）、DEL（删除）等键。

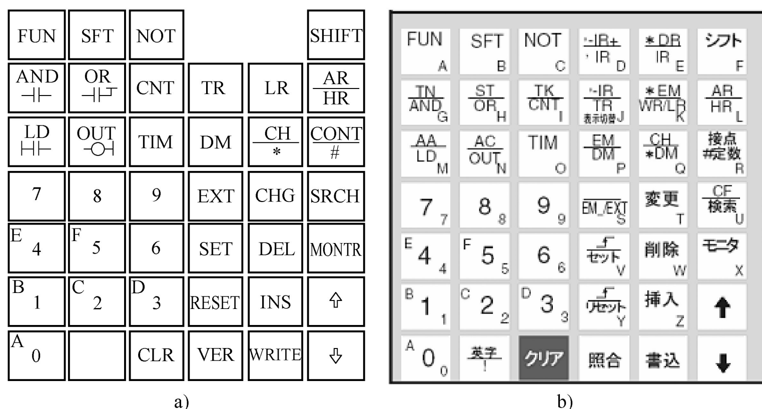


图 2-80　简易编程器键盘

2. 操作

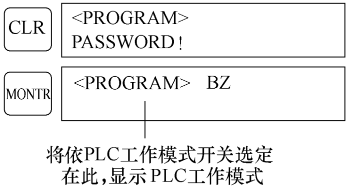
可用编程器作多种操作。

（1）预备性操作。有：

输入口令，使编程器联机工作：编程器接上 PLC，PLC 接上电源后（编程器电源由 PLC 支持），编程器显示屏上会出现如下字样：

```
<PROGRAM>
PASSWORD!
```

这时应输入口令。系统提供的口令为，依次按 CLR、MONTR 两个键。



输入后，将依 PLC 工作模式开关选定，显示 PLC 工作模式。

PLC 中存的用户程序，如加有密码，须先解密。

提示：解密的步骤是，先把液晶显示清零（按“CLEAR”键），然后按“SHIFT”、“CLEAR”键，再按“RESET”键。这时，显示屏显示4个问号，要求送入密码。送入的密码后，再按“WRITE”键，予以确认。

如送入的密码正确，即可读出程序，否则，只是显示地址 0000，什么也读不出来，进一步操作也无法进行。

蜂鸣器：编程器还有蜂鸣器，其是否发声可设定。办法是，先依次键入“CLEAR”、“SHIFT”、“CLEAR”键，将显示蜂鸣器工作模式。如显示屏显示为如下所示，有“BZ”字样，则蜂鸣器会发声。

```
<MONITOR>   BZ
         . . . . . . . . . . .
```

当蜂鸣器会发声时，若键入指令，数据，键入出错时，蜂鸣器将发“的”或“的的”声。如不要其发声，可先键入“SHIFT”键，再“1”键，显示屏将不显示有“BZ”字样显示，即：

```
<MONITOR>
         . . . . . . . . . . .
```

则不会发声。如再想发声，还是先键入“SHIFT”键，再“1”键，其显示又回到原来状态，又能发声。

声音的大小，可用编程器上方侧面的一个电位器旋钮调整。

清零：依次按 CLR、PLAY/SET、NOT 和 REC/RESET，这时显示屏出现：

```
00000MEMORY CLR?
   HR CNT DM
```

若再按 MONTR 键，显示：

```
00000MEMORY CLR
END HR CNT DM
```

表示用户程序全部被清除，而且 HR、CNT、DM 也清零。

如果在按 MONTR 之前，先按 HR、CNT、DM 中任一个，或两个、三个，再按 MONTR，则相应的 HR、CNT 或 DM 不被清零。

建立地址：按 CLR 键，显示 0000。未显示时还可多按几次，直到显示 0000。这时建立的地址，即为

0000。如想建立另一个地址，可依所要建立的地址号键入。

（2）编程。这是最基本的操作，应在编程状态下进行。有：

写入指令：写入指令先是地址（从零开始，从小到大，自动生成），后操作码，再操作数，最后为 WRITE 键。功能指令则要先按功能键，后再按功能号（两位或三位数）。

读出指令：编程过程常要把已送入的指令读出，以了解编程的情况。这时可先送地址，后按下移键即可。若再想读下一条指令，可再按下移键。想读上一条指令，也可按上移键。

插入指令：先找出所要插的位置（指插在它之前），办法是依上述读出指令的方法读出它。然后写入要插的指令，再按插入键，最后按下移键。

删除指令：先找出所要删的指令，再按删除键，再按上移键。

程序检查：先清零，再按 SRCH 键，即可进行检查。如程序有错，即可显示错误指令的地址，及错误号。如没有错，则显示最末一条指令地址及 END 字样。

（3）监控。编程器不仅可用于编程，还可用作对 PLC 的监控。其操作为

1）查找指令：清零，键入指令，再按 SRCH 键，即显示查找的指令及其地址。

2）查找触点：清零，按 SHIFT 键，再按 CONT/#键，再按要找的触点号，再按 SRCH 键。这时将显示第一个使用该触点的地址及指令。再按 SRCH，还会出现后续使用该触点的地址及指令。

3）读出及清除出错信息：PLC 出现非致命错误（NON-ERROR）可用编程器显示与清除。其操作为：先按“CLEAR”键，显示屏清零；再按“MONTR”键。如有错，则显示有关出错信息。如有多个错误，多次按“MONTR”键，可依次显示这多个错误。

有了错误，要查出错原因，并予以清除。同时，还须清除出错信息。否则 PLC 的有关报警灯还闪亮。清除的方法是，先按“CLEAR”键，显示屏清零；再按“FUN”及“MONTR”键。如所有错误已排除，则显示“ERR/MSG CHK OK”。

4）数据监视：查出触点或指令后，按 MONTR 键，即可监视触点 ON、OFF 的情况。如监视的为定时器或计数器，还可显示它的当前值。再按上移或下移键，还可显示相邻触点号的触点 ON、OFF 情况，也可显示通道或数据存储器的内容。另外，还可作多点显示，即显示一个点后，还可再送入点进行监视。总共可显示 6 点，但只有 3 点出现在显示屏上，按 MONTR 键可把未显示的点滚动到显示屏上，而最先显示的从显示屏上消失。

5）微分监控：用以检测某位从 OFF 到 ON，或从 ON 到 OFF 的信息。如检测到此信息，将发声，并在显示屏上有相应显示。如有下图的位数据监控例子：

```
L000000001H0000
^OFF^OFF ^OFF
```

这时同时监控 3 个位，L0000、00001 及 H0000。但 L0000 位在显示屏的最左方，故可对其作微分监控设定。如为如下键入：

```
[SHIFT] [↓]    L000000001H0000
               U@OFF^OFF^OFF
```

为 L0000 的上升沿微分监控，当检测到 L0000 从 OFF 到 ON 时，蜂鸣器将发出声音。

如为如下键入：

```
[SHIFT] [↓]    L000000001H0000
               D@OFF^OFF^OFF
```

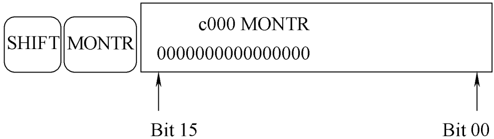
为 L0000 的下降沿微分监控，当检测到 L0000 从 ON 到 OFF 时，蜂鸣器将发出声音。

如要监控别的位，则先按相应的若干次“MONTR”键，把要监控的位地址显示在显示屏的最左方，再作与以上相同的设定即可。

6）二进制监控：如被监控的字处于显示屏的最左方，如下图所示。

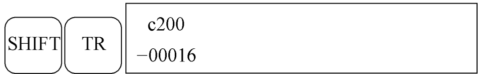


这时，将监控 000 通道的值。如作了如下键入，则可实现二进制监控。

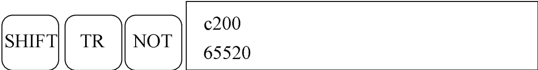


这时，将按二进制的格式显示 000 通道的值。这时，如再键入“CLEAR”键，又将回到对通道 000 的正常显示。

如要用带符号位的显示通道的值，则可作如下键入。



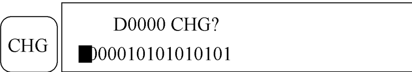
这时，如再键入“CLEAR”键，又将回到对通道 000 的正常显示。如要用不带符号位的显示通道的值，则可作如下键入。



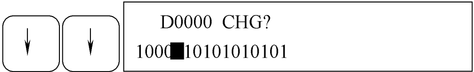
这时，如再键入“CLEAR”键，又将回到对通道 000 的正常显示。

7）设定或强制 ON/OFF：监视到触点的 ON、OFF，还可对其状态做改变或强制改变。OFF 改变成 ON，按 PLAY/SET 键；强制 ON，按 SHIFT 及 PLAY/SET 键。改变成 OFF，按 REC/RESET 键；强制成 OFF，按 SHIFT 及 REC/RESET 键。强制定时器、计数器时，它的当前值也会作相应改变。

当作了二进制监控设定后，也可对相应的位，作 1 或 0 的设定或强制。如作了如下键入，

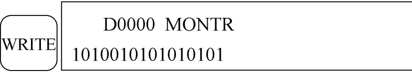


则有光标出现。上图光标显示在第 15 位。如要对别的位操作，则可相应地键入“↑”或“↓”，这将改变光标位置：



光标位置选定后，如要对该位置 1，则键入“1”。如要对该位置 0，则键入“0”。如要对该位强制 1，则键入“SHIFT”及“SET”。如要对该位强制 0，则键入“SHIFT”及“RESET”。如要取消对该位强制，则键入“NOT”。

作了以上键入后，再键入“WRITE”，J 将把这个改变写入 PLC 内存，显示屏则又回到二进制监控时的显示。即：



8）改变 T/C 的设定值：查出所要改设定值的指令，按 CHG 键，然后再送入新值，再按 WRITE 键即可。

9）改变 T/C 的当前值：查出所要改当前值的指令，按 MONTR 键，对其监视，再按 CHG 键。然后送新值，再按 WRITE 键。当然，这时 PLC 应处监控状态。

10）改变 DM 当前值：键入要改的某 DM 地址，按 MONTR 键，对其监视，再按 CHG 键。然后送新值，

再按 WRITE 键。

11）读扫描时间：清零，再按 MONTR 键，显示屏即显示 PLC 执行当前程序的时间。

以上操作多须在 MONITOR 状态下进行。查找、监视、读扫描时间也可在 RUN 状态下进行。查找还可在编程状态进行。

（4）存储程序：编程器上可接入磁带录音机，可把程序存入磁带；也可从磁带上读出程序，并存于 PLC 中。

在向磁带存程序，或从磁带读程序时，PLC 必须处于编程状态。普通的录音机即可用于存储，半小时的磁带可存 16KB 的程序。

不过，由于编程软件的完善及个人计算机使用的普及，存贮程序多用计算机的外设实现，已不用这个磁带机了。所以，对这个程序存贮与调入的过程就不加介绍了。

（5）I/O 表登记

箱体式 PLC（含 CQM1、CQM1H），不须定义 I/O 表，也没有 I/O 表登记。它的 I/O 地址与箱体间连接的情况有关。依据它们之间的关系，弄清各个 I/O 地址也就可以编程了。

模块式 PLC，要用 I/O 表定义 PLC 所安装的机架与单元（模块）的地址。显然，I/O 表没定义好，PLC 的 I/O 地址不确定，是无法编程的。

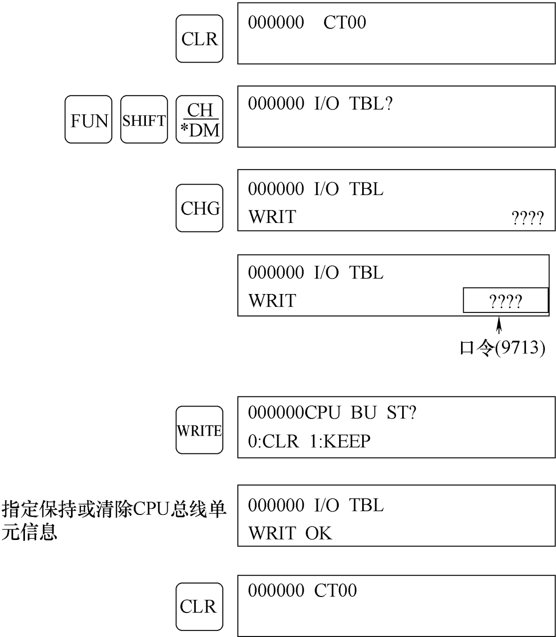
只是，OMRON 公司模块式 PLC 的 I/O 表可自动形成，无须设计。这时，其 I/O 地址将按默认确定。

自动形成的 I/O 表可作登记。一经 I/O 表登记，PLC 运行前，CPU 就要检查实际模块连接与 I/O 表是否相符。如不符，则出现 I/O 总线错，PLC 无法进入运行模式，无法工作。

自动形成的 I/O 表也可不作登记。这样做，当 PLC 上电时，其 CPU 不检查实际模块连接与 I/O 表（因未登记，无 I/O 表）是否相符。不管模块是如何安装的，其程序照样运行。这当然有一定的危险。所以，一般还是推荐作 I/O 表登记。

提示：用简易编程器 I/O 表登记是很方便的。请记住如下步骤。

I/O 表登记可用编程软件（见后），也可用简易编程器。把简易编程器连接到 PLC 上，PLC 上电，并置于编程模式。然后，再按如下步骤操作即可。



如为 CS1/CJ1 机型前生产的 PLC，则按“WRITE”键后，“指定保持或清除 CPU 总线单元信息”框不会出现，不必选择键入“0”或“1”。如无其它问题，将直接出现“WRITE OK”框。

提示：登记后的 I/O 表也可清除。其步骤只是在以上步骤中，在键入“WRITE”键之前，加入“NOT”键。

（6）PLC 设定

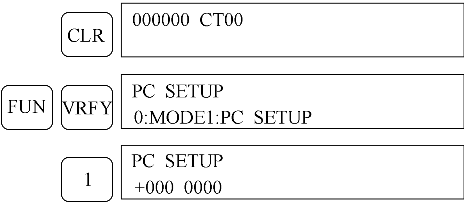
按要求，对 PLC 设定区的内存内容写入相应的值，称 CPU 软设定。这是使 PLC 按要求进行工作，对 PLC 进行初始操作的重要内容。

用编程器作这个设定是很方便的。作 PLC 设定时，可选择相应字地址，并写入相应值实现。如 CJ1 机，其监视循环时间（以 10ms 为单位）的设定地址为 209。具体见表 2-24。

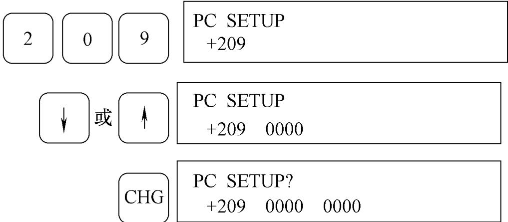
表 2-24　监视循环时间设置

地　址	位	设　置	设 置 范 围
209	15	允许监视循环时间设置	0：使用默认设置 1：使用 0～14 位的设置
	0～14	监视循环时间设置	0001～0FA0

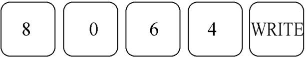
对此，如用编程器作设定，首先，编程器与 PLC 联机，使 PLC 处编程模式。然后，选定进行 PLC 设定。即：



进而，选择地址 209，并按“CHG”键。即：



再键入“8”、“0”、“6”、“4“、“WRITE”，即：



则出现：



这样，即完成了对监视循环时间的设定。本例第 15 位为 1，故监视循环时间为第 14 位到第 0 位间的值（应转换为十进制）及 10ms 乘积，即 1s。

如为 CS1/CJ1 之前的 PLC，可直接对 DM 的设定区，写有关数据。

当然，要做这类设定，首先要弄清 PLC 设定的有关地址、要求，否则，这类设定是无法进行的。至于设定的有关地址、要求，请参阅有关说明书。

编程器还可作其它操作，并依所面对的 PLC 的型号不同而有所不同。这里就不再赘述。

2.5.2　图形编程器

简易编程器只能用助记符编程，显示屏小，功能有限。多数公司都开发有图形编程器。图 2-81 所示为 OMRON 公司的图形编程器，类似笔记本电脑。

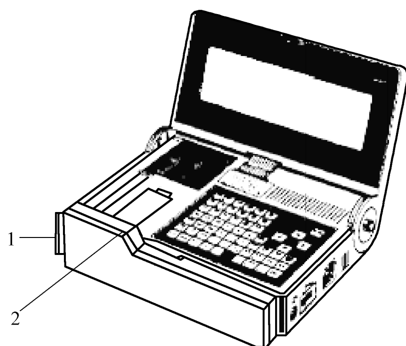


图 2-81　图形编程器
1—内存卡　2—内存卡指示器

图形编程器既可用助记符语言编程，还支持梯形图语言编程。而且，它的外设多，可用其打印程序、数据，如图 2-82 所示。还可用其在软盘上存储文件，如图 2-83 所示。

但它的价格非常贵，加上近年来个人计算机飞速发展及 PLC 编程软件不断完善，图形编程器已被“笔记本电脑加软件”取代。所以，这里不再进一步对其作介绍了。

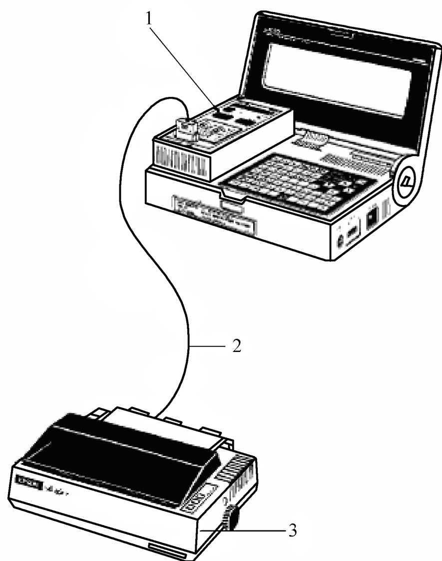


图 2-82　图形编程器接打印机
1—打印机接口单元　2—打印机电缆　3—打印机

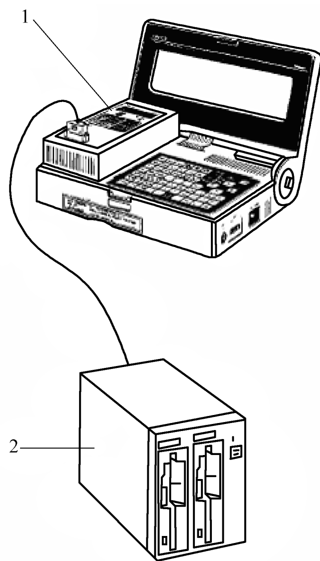


图 2-83　图形编程器接软盘驱动器
1—软盘接口单元　2—软盘驱动器

2.6　PLC 编程软件

关键词：CX-Programmer、窗口、下拉菜单、弹出菜单、工具条、快捷键、脱机、联机、属性设定

2.6.1　概述

OMRON PLC 编程软件为 CX-One，基于 CPS（Component and Network Profile Sheet）集成开发环境。其主要有：CX-Programmer（用于编程）、CX-Simulator（用于编程仿真）、NS-Designer（用于可编程终端编程）、CX-Motion（用于运动控制编程）、CX-Protocol（用于协议宏通信编程）、CX-Process Tool（用于模拟量控制编程）、CX-Server（用于网络配置与管

理），等等。最近已升级到 7.2 版本。新版本可支持最新的机型。同时可安装在 VISTA 的平台上。

图 2-84 所示的为 CX-One 安装后出现在 Windows Vista 后，“开始”、“OMRON”菜单下的项目。

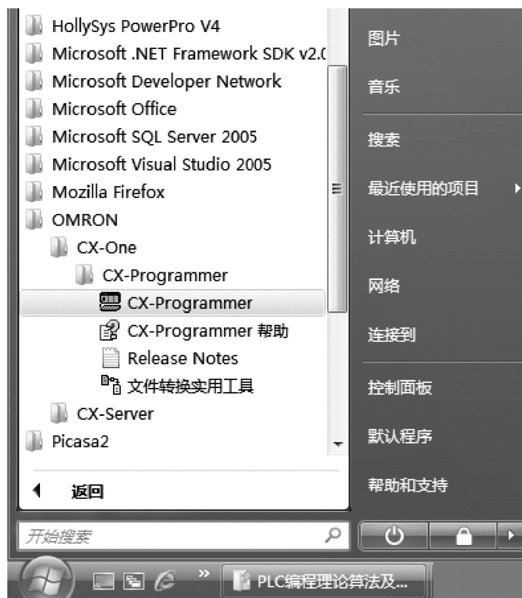


图 2-84 CX-One 安装后出现在 Vista“所有程序”菜单下的项目

编程软件主要功能是：

1）编程：可用其选用多种语言，编写 PLC 程序，并可进行相关语法检查；

2）上、下载程序：可用其把已编写好的程序下载给 PLC，或从 PLC 上读取已有程序；

3）监控：可用其监控 PLC 工作，包括回路监视，软元件同时监视，软元件登录监视等。

4）调试：可用其测试在 PLC 上的程序能否正常运行。也可使用仿真软件，在计算机对上，对所编的程序进行仿真调试。

5）PC 诊断：可用其诊断 PLC 状态，查找其故障履历。

6）设定：可对 PLC 进行种种设定，以确保其能正确工作。

……

提示：在 OMRON 官方网站上，可免费下载 CX-one7.0简化版本。可用于 OMRON 小型机（包括 CP1L 等新型机）编程、监控。

2.6.2 组成

CX-Programmer 用的是完全视窗风格的界面。有窗口、菜单、工具条、状态条。可用鼠标操作与键盘操作。并可打开多个例程（Instance），多个窗口，多个 PLC，多个程序，多个（程序）段进行处理。

1. 窗口

CX-Programmer 用的窗口也有 3 种：重叠（Overlapped）窗口，弹出（Popup）窗口和子窗口（Child）。

重叠窗口通常用于建立应用程序主窗口。有时也叫父窗口或称“框架”窗口。图 2-85 所示为 CX-Programmer 的父窗口。它也是打开本软件后首先出现的窗口。

子窗口显现在框架（父）窗口的用户工作区内。只有打开或新建文件后才可能出现子窗口。子窗口一般也只能在框架（父）窗口显示与移动。

CX-Programmer 的子窗口有主（子）窗口及辅（子）窗口两类。

主（子）窗口有 5 个，分别用以显示梯形图，助记符，全局符号，局部符号及交叉引用数据的画面，可相应进行梯形图、助记符、全局符号、局部符号编辑以及察看变量交叉引用的情况。

辅助（子）窗口有 3 个，分别为工程工作区窗口，输出窗口及观察窗口。

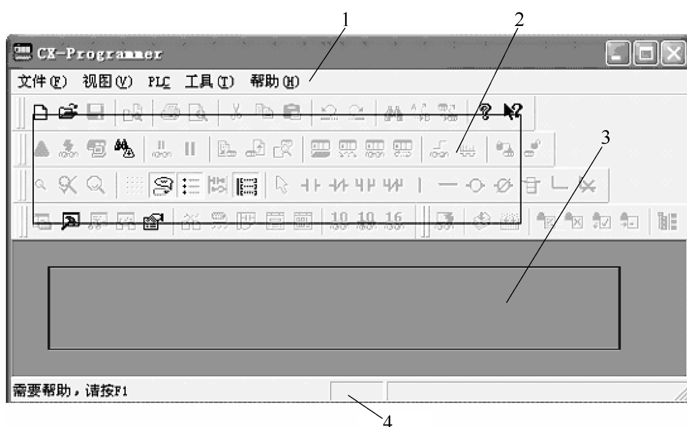


图 2-85　CX-Programmer 父窗口

1—菜单　2—工具条　3—用户工作区　4—状态条

图 2-86 所示就显示了主窗口（显示梯形图）及 3 个辅窗口的画面。

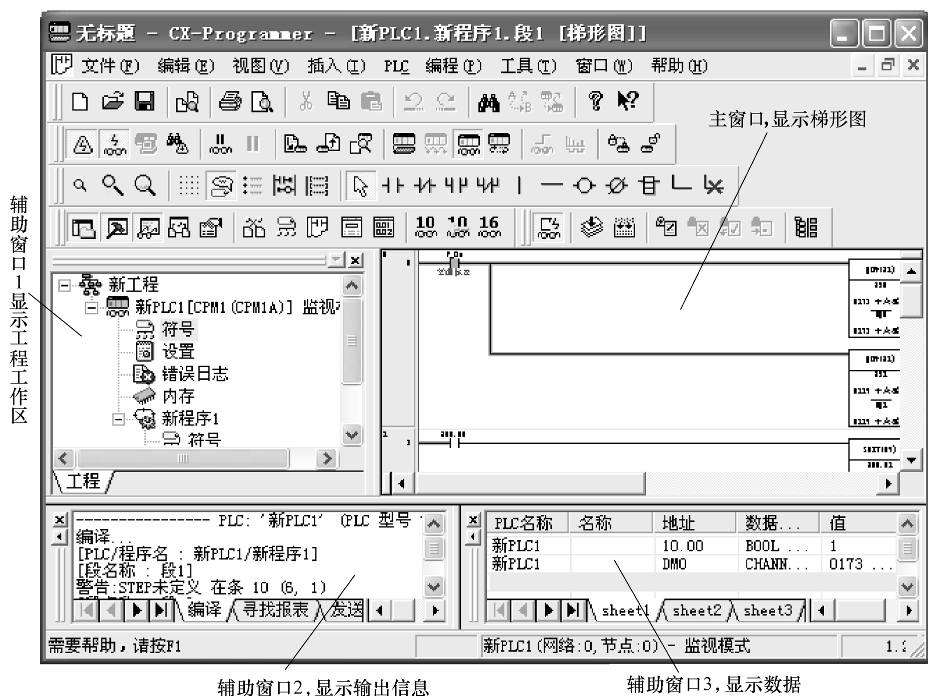


图 2-86　主窗口及 3 个辅助窗口

辅助窗口可打开，也可关闭。这可用鼠标点击响应菜单项实现。也可用热键操作。可分别用“ALT 键 +1 键”打开或关闭工程工作区窗口，用“ALT 键 +2 键”打开或关闭输出窗口，可用“ALT 键 +3 键”打开或关闭观察窗口。如窗口打开，此操作后，则关闭；反之，则打开。

工程工作区窗口相当于目录窗口，它的目录项可打（展）开，也可缩回，如同其它 Windows 类似界面一样。而且，在它的目录项打开后，双击其中的任意项，即可弹出相应的画面。

打开的主窗口，可在整个用户工作区内显示。如打开的主窗口较多，可层叠显示，也可

平铺显示。图 2-87 所示为层叠显示。但辅助窗口设为浮动时，也可占满整个用户工作区，也可与主窗口一样处理。

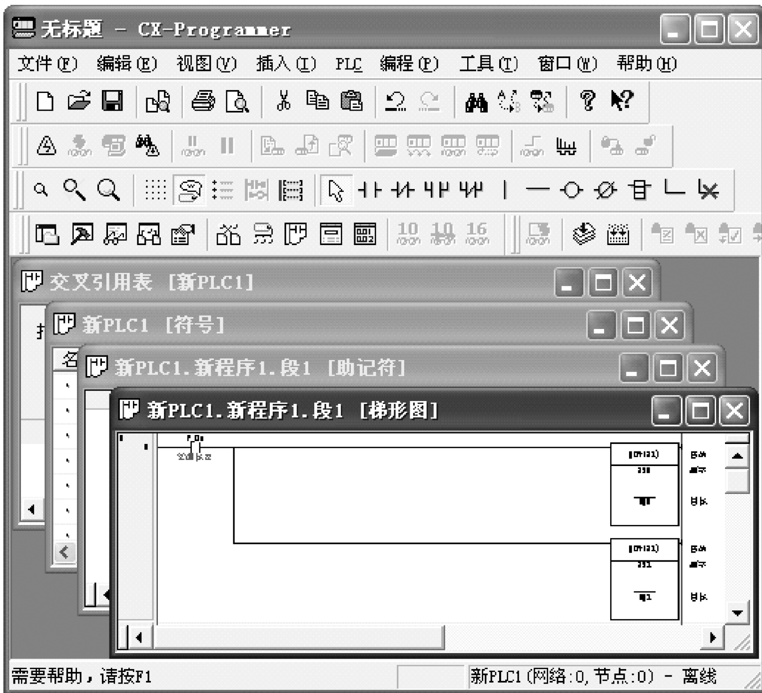


图 2-87　窗口层叠显示

图 2-88 所示为平铺显示。

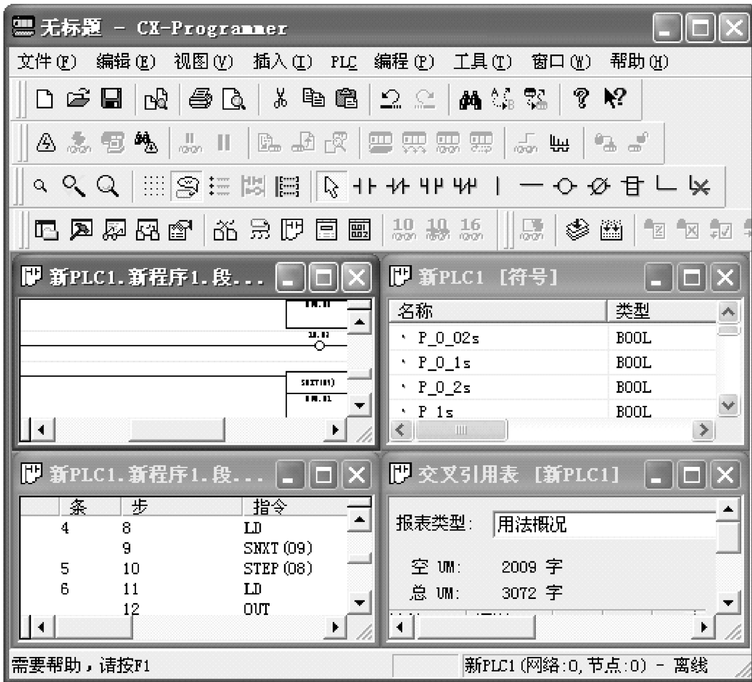


图 2-88　窗口平铺显示

显示梯形图的窗口还可进行分割，可分为四份或两份。如图 2-89 所示。

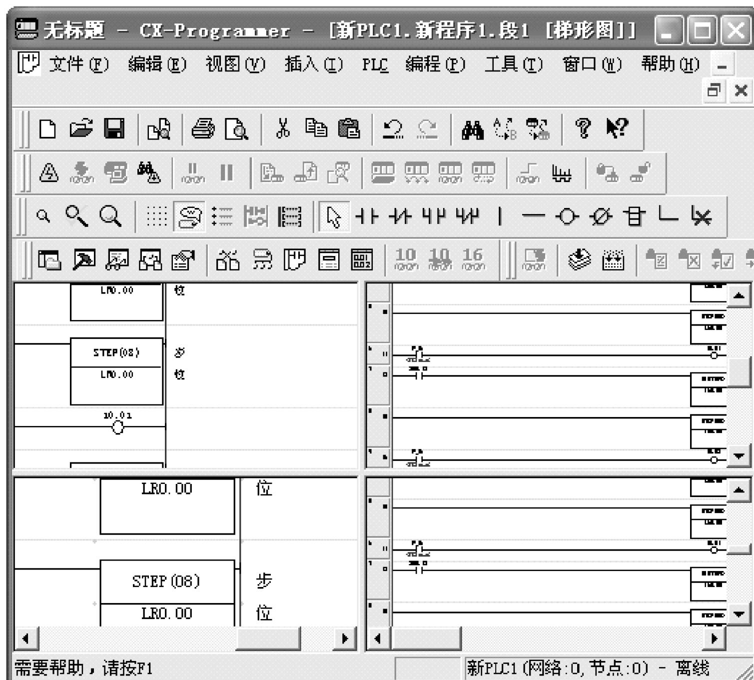


图 2-89　窗口分割显示

如图所示，分割后的四个部分，可显示相同程序的不同部分的内容，以便于同时对程序的不同部分进行编辑与观察。

弹出窗口也称对话窗口，用以显示信息或与用户对话。

图 2-90 所示为一对话窗口。它是建立新建文件时出现的，要求使用者作出相应的选择。

这里的设定与设置击后，还会弹出相应的对话窗口，使用者还可进一步作出相应的选择。图 2-91 所示为点击这里的设定后又出现的窗口。



图 2-90　对话窗口

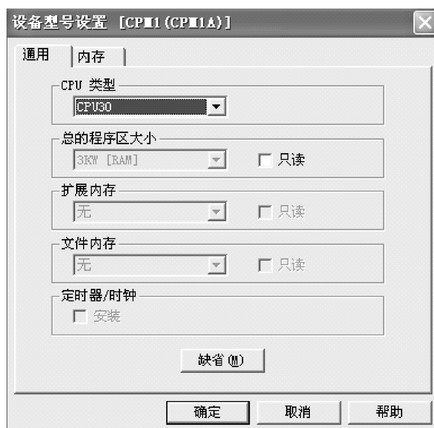


图 2-91　又一对话窗口

CX-one 软件还有含有若干工具软件（CX_Server）及其相应工作窗口：内存窗口（PLC Memory Component），PLC I/O 表窗口（IO Table Component），PLC 设定窗口（PLC Setup

Component)，数据跟踪/时间图图监控窗口（Data Trace/Time Chart Monitor Component)，内存卡窗口（Memory Card Component)，网络管理窗口（CX-Server（CX-Net）Network Configuration tool）及 PLC 时钟工具窗口（PLC Clock Tool）等。这些窗口有的也是父子式的，在框架窗口内也可有很多子窗口。

图 2-92 所示为内存窗口，用以向 PLC 读写数据。它的风格与本主体窗口风格是基本一样的。

图 2-93 所示为时间图监控窗口，可用时间图的方式从 PLC 读取数据。可用它建很多子窗口，以实现种种数据采集，并用图形显示。

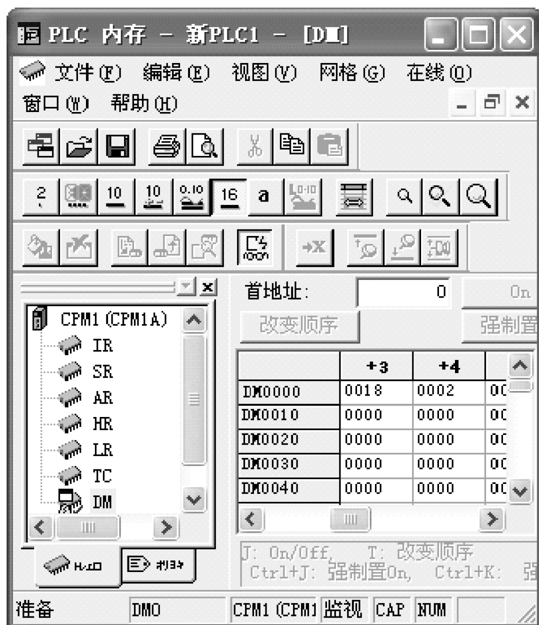


图 2-92　内存窗口

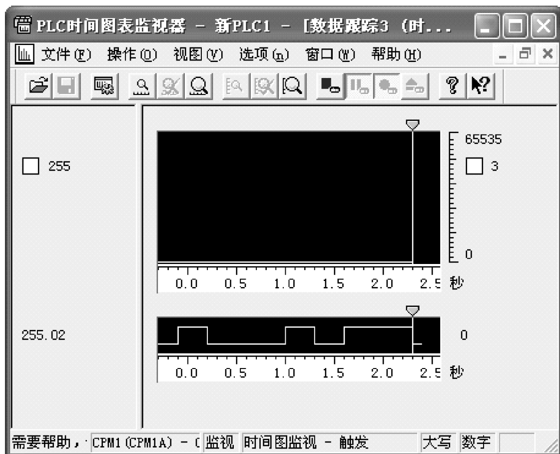


图 2-93　时间图监控窗口

这些窗口，在工程工作区中点击相应项，或点击 CX-Programmer“框架”窗口的相应菜单项，或点击工具条上的相应项，即可出现，并独立于 CX-Programmer 的“框架”窗口。但当退出 CX-Programmer 后，这些窗口也消失。

2. 菜单

本系统用的菜单有两种：下拉菜单与弹出菜单。

(1) 下拉菜单。依不同窗口的打开而有所变化。当打开或新建文件后，主下拉菜单项有：文件、编辑、示图、插入、PLC、编程、工具、窗口、帮助 9 项。

这里的文件、编辑、示图、窗口、帮助基本与标准的 Windows 的界面是相同的。而插入、PLC、编程 3 项是本软件特有的。

1）插入项。如工程工作区击活，可用以插入 PLC，插入程序，插入段；如符号画面击活，可用以插入符号。其子菜单如图 2-94 所示。

如梯形图画面可用以插入梯形图符号等，其子菜单如图 2-95 所示。

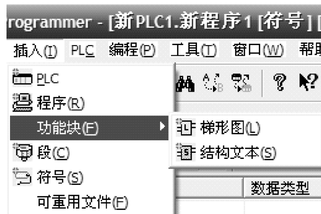


图 2-94　下拉菜单

2）PLC 项。其子菜单项较多，如图 2-96 所示。

3）编程项。图 2-97 所示为它的展开菜单。它用于程序编译与在线编辑。

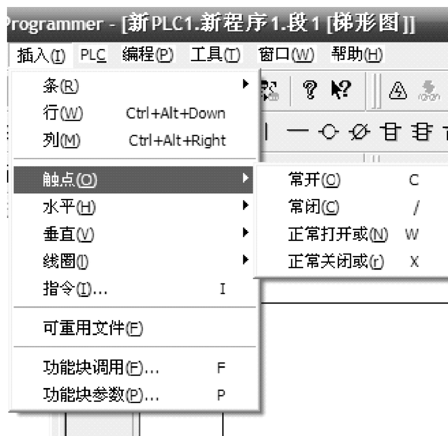


图 2-95　插入子菜单

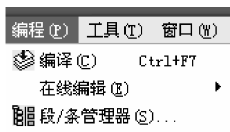


图 2-97　编程项



图 2-96　子菜单项

（2）弹出菜单。在不同窗口，不同位置，右击鼠标时，多会弹出一个菜单。此即弹出菜单。所弹出菜单的内容，依右击鼠标时所在的窗口或位置不同而有所不同。图 2-98 所示为在工程工作区右击鼠标时弹出的一个菜单。

图 2-99 所示为在工程工作区新 PLC［CS1G］处右击鼠标时弹出的一个菜单。

图 2-100 为在工程工作区新 PLC［CS1G］处右击鼠标时弹出的一个菜单。

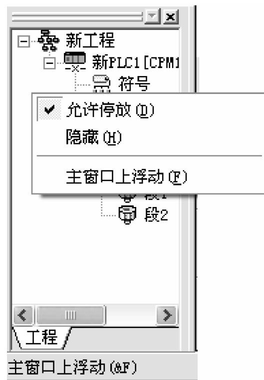


图 2-98　弹出菜单



图 2-99　又一弹出菜单 1

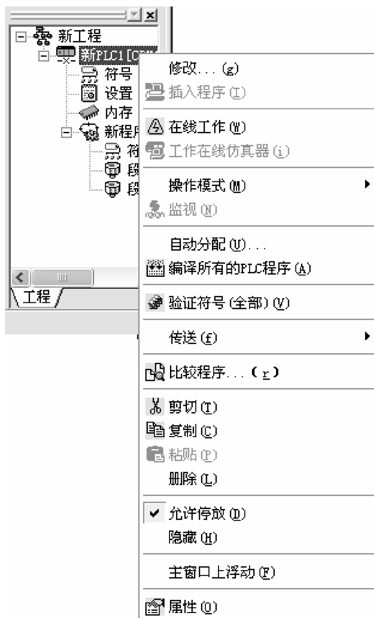


图 2-100　又一弹出菜单 2

等等。

在弹出菜单出现后，再左击鼠标左键，即可进入相应的操作。

3. 工具条

共有 7 个工具条。它们是：梯形图、PLC、标准、查看、程序、插入及符号表。图 2-101为这 7 个工具条，只是它未放在窗口的顶部。

这 7 个工具条的功能及其是否激活与相应的菜单项是对应的。其具体的情况介绍如下：

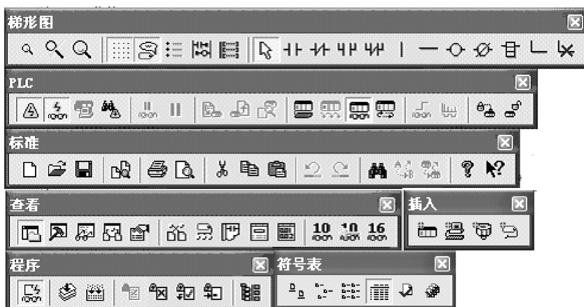


图 2-101　7 个工具条

标准工具条：它为 Windows 界面通用。有 16 项，含有文件操作、文本编辑等功能，与其它的 Windows 界面类似工具条相同。

梯形图工具条：有 20 项。用以梯形图编辑。

查看工具条：有 13 项，用于显示窗口的选择。

PLC 工具条：有 17 项，用以选择与 PLC 通信，如联机，脱机；监控，不监控；下载，上载；微分监控，数据跟纵或时间图监视器；加密，解密等。

程序工具条：有 8 项，用以选择监控，程序编译，程序在线编辑。

对工具条操作比对菜单项操作功能相同，但要简单与迅速。所以，编程人员要努力学会操作它。

插入工具条：有 4 项，用以进行新 PLC、新程序、新段、新符号插入操作。

符号表工具条：有 6 项，用以进行符号表显示或检查操作。

4. 状态条

显示在窗口的最下方，用以提示有关状态信息。

2.6.3　操作

本软件可用鼠标或键盘进行操作。

1. 鼠标

如同其它 Windows 界面，鼠标有 4 种操作：左单击、左双击、右单击、按左键时拖放。在不同窗口或不同的项目或不同的画面下，对这 4 种操作作些测试，即可得知这些操作各有什么功能。

2. 键盘

用以对系统操作及输入数据。对系统操作则用热键。输入数据按提示进行。用热键操作再与输入数据结合，速度快，是提高编程效率所必须要做的。以用梯图进行编程为例，可知用热键比用鼠标是要快得多的。如移动光标可用方向键，它不比鼠标慢。光标在合适位置时，如插入一行用 CTRL + ALT + 向下键要比用鼠标操作快。光标在合适位置时，如插入一列用 CTRL + ALT + 向左键也要比用鼠标操作快。光标在合适位置时，如删除一行用 CTRL + ALT + 向上键要比用鼠标操作快。光标在合适位置时，如删除一列用 CTRL + ALT + 向右键要比用鼠标操作快。

插入指令也一样。如用 ALT + I，再用 ALT + O，再 ALT + O 比用鼠标插入常开节点也快得多。如用 ALT + I，再用 ALT + L，再 ALT + O 比用鼠标插入常开线圈也快得多。如用 ALT + I，再用 ALT + I 比用鼠标插入指令也快得多。如用 CTRL + F7 比用鼠标进行程序编译也快得多。

等等。

还应指出的是，CX-Programmer 用户可依需要，自身定义热键。这可在“工具”菜单项中，再击“键盘映射”项，即可显现图 2-102 所示的“快捷键”窗口。

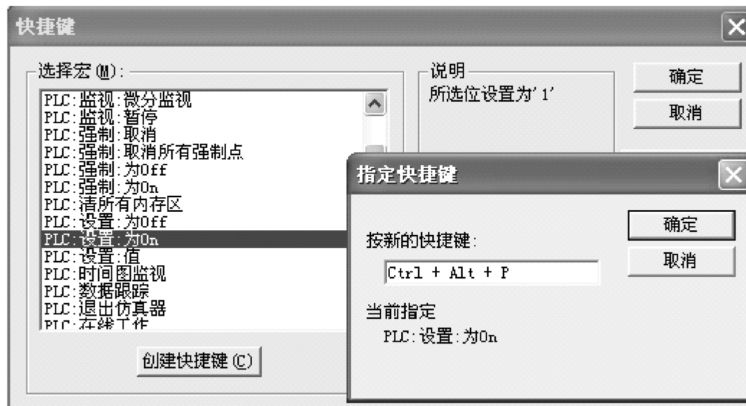


图 2-102　快捷键窗口

如图所示，这里先在“选择宏”中，选中“PLC：设置：为 ON”项，然后，击“创建快捷键”按钮，这时，将弹出“指定快捷键”对话框。进而，同时按下“Ctrl”，“Alt”及“p”3 个键（如选用这 3 个键同时按下，作“PLC：设置：为 ON”快捷键），则在对话框的“按新的快捷键”栏处，则出现“Ctrl + Alt + p”字样。进一步，击对话框的“确定”键，对话框消失，再击“快捷键”窗口的”确定”键，“快捷键窗口消失”。到此，即完成热键的自定义。

完成这个自定义后，如进入监控（含义见后）状态，在梯形图上选中某触点后，再同时按下“Ctrl”、“Alt”及“p”3 个键，则可使该节点置“ON”。

选择宏的项目很多，表上的所有项目，均可对其自定义一个快捷键。很明显，依需要自定义了一些快捷键，然后，再用快捷键作相应操作，定将提高使用 CX-Programmer 软件的效率，并可简化 CX-Programmer 软件的操作。

2.6.4　使用

本软件主要在两种状态下使用：离线（脱机）与在线（联机）。

1. 离线

即脱机，主要是编程。具体工作是：PLC 设定、I/O 地址分配与变量编辑以及用梯形图（或其它语言）编程。

（1）PLC 配置与设定。又称组态，主要是利用相应窗口，做好如下几项设定：

1）选择 PLC 型号及 CPU 版本。

2）根据 PLC 的硬件配置，作好各个硬件单元或模块的设定。硬件设定有的是用模块上的设定开关，有的用编程软件，有的两者都用。

3）根据需要，对 PLC 的内部器件及有关参数做好设定。

4）根据联网情况，作好联机通信的有关设定。

5）如需要作程序加密设定。

以上设定可用鼠标点击不同菜单项或工具条，选择相应的对话窗口实现。尽管不同的 PLC 及其硬件配置，要做的这些设定差别很大，但都必须用编程软件一一完成。

（2）对象属性。CX-programmer 是面向对象的软件。每一窗口或对象（Object）一般都有与其相关属性（Property）。可选定的对象有：工程（Project）、PLC、程序（Program）及段（Section）。

查看对象属性的方法为

1）从工程工作区窗口选定要察看的对象；

2）击鼠标右键，等待弹出菜单；

3）从弹出菜单中，选属性项，并点击之；

4）有关对象的属性项对话窗口将显示。

图 2-103 所示为工程属性窗口。用其可定义工程名称，增加注释，并可用其链接 CX-SERVER 文件，以达到与别的工程实现数据共享。

图 2-104 为 PLC 属性窗口。用其可定义 PLC 名称，并对一些编程中的重要特性作设定。如“以二进制形式执行定时器/计数器”项，若作选定（右方小窗口打勾），则可起用 TIMX 等以二进制形式执行的定时器/计数器指令。若未作选定（右方小窗口未打勾），则只能起用 TIM 等以 BCD 码形式执行的定时器/计数器指令。

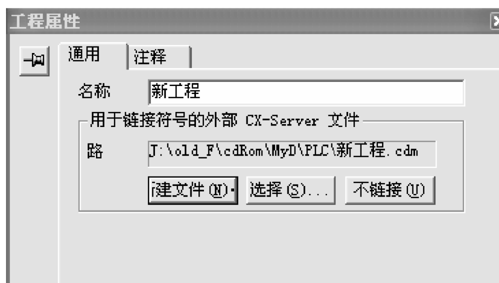


图 2-103　工程属性窗口

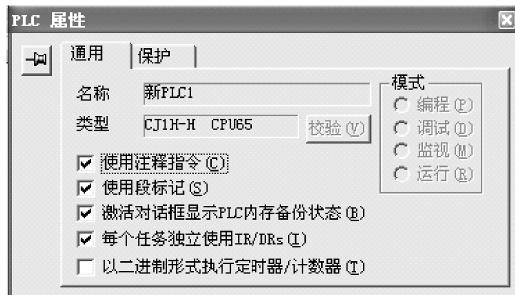


图 2-104　PLC 属性窗口

在该窗口上，如单击“保护”标签，将出现密码设定对话框。在其上可键入程序保护的密码。密码为 8 位数，含英文字母或数字。

提示：这里用的加密方法只适用于 CS、CJ 等 OMRON 新型 PLC。

当然，这个 PLC 属性窗口与所用的 PLC 有关。本例的 PLC 为 CJ1H 机，才有那么多设定选项。如为其它机型，可能就没有那么多项。如 CPM1A，选项很少，也不能用这种方法对程序加密。



图 2-105　程序属性窗口

图 2-105 所示为程序属性窗口。可定义程序名称，增加注释，程序类型（循环

或中断）及操作是否在 PLC 进入运行模式时，就开始执行。

当然，有的 PLC 的程序属性，较简单，不一定就有这些选项。

作为对象的段也有相应属性。出现它的属性窗口后，也可用其定义段的名称及增加注释。

图 2-106 所示为 CX-Programmer 的一个设定窗口。其上有多个表单，可根据表单的内容作相应设定。该图显示的为 CPU 设置表单。



图 2-106　CX-Programmer 的一个设定窗口

提示：对 PLC 进行设定，往往不只用一个对话窗口就可完成，特别是中、大型机。如图2-103 ~图2-105，要细心查找有关菜单项，全面完成设定工作。

（3）I/O 表设计。模块式 PLC 的 I/O 表，可自动形成，也可自行设计。自动形成时，其 I/O 地址按默认值确定。自行设计时有的 PLC，如 CJ1 机，其地址可按给定的变化范围选定，较灵活。

要自行设计时，可双击工程工作区中的 I/O 表项，将弹出 I/O 表设计窗口。该窗口提供了可能的 I/O 配置。你可按你的系统实际配置进行选择。

设计后，再传送给 PLC（PLC 应与计算机联机，且 PLC 处编程模式）。这也就完成了 I/O登记。一经 I/O 表登记，PLC 运行前 ，CPU 就要检查实际模块连接与 I/O 表是否相符。如不符，则出现 I/O 确认错，PLC 无法进入运行模式，无法工作。

自动形成的 I/O 表也可作登记。登记时，首先，PLC 与计算机联机，且使 PLC 处编程模式。进而，双击工程工作区中的 I/O 表项，等待弹出 I/O 表设计窗口。该窗口出现后，再在其上的“选项”菜单项中，选“创建项”项，并单击之。

自动形成的 I/O 表也可作不登记。这样做，当 PLC 上电时，其 CPU 不检查实际模块连接与 I/O 表（因未登记，无 I/O 表）是否相符。不管模块是如何安装的，其程序照样运行。这当然有一定的危险。所以，一般还是推荐作 I/O 表登记。

I/O 表登记可用 CX-Programmer 软件删除。办法是：PLC 与计算机联机，且使 PLC 处编程模式。双击工程工作区中的 I/O 表项，等待弹出 I/O 表设计窗口。该窗口出现后，再在其上的“选项”菜单项中，选“删除”项，并单击之。

对 CJ1 机，I/O 是否登记可在辅助区 A260 中进行检查。如果 A260 内容为 0000（十六进制），则 I/O 表未登记；PLC 每次上电时，根据 PLC 实际连接的模块（单元）创建 I/O 表。如果 A260 内容为 BBBB（十六进制），则 I/O 表已登记；PLC 每次上电时，CPU 将根据已登记的 I/O 表，检查中实际连接的 I/O 模块（单元）。如实际连接的 I/O 模块（单元）与登记的 I/O 表不符，则将有 I/O 确认出错显示，PLC 将不能工作。

提示：I/O 表登记后，如实际安装模块状况与登记的状况不同，PLC 将出现致命错误，只能处于编程模式，不能进入监控或运行模式。

（4）符号编辑。本软件允许用符号，即变量，为 I/O 或内部器件的地址名。用它代替 PLC I/O 或其它内部器件的地址。如按实际内容命符号名，可为程序读、写，提供了方便。同时，用符号编的程序，可作到与地址无关，实现标准化。改用时，符号与地址重新作对应，也就可以了。

符号变量有全局与局部两种。

全局变量在所选的 PLC 内有效。而局部变量仅在所编的程序中有效。

这些变量编辑可在相应的符号编辑窗口中进行。

提示：老式 PLC，下载程序时，这里的符号不能下载到 PLC 中。PLC 保存的只是梯图程序编译后机器码，不保存符号。所以，从 PLC 上载的程序用的是地址，而不是符号。

（5）梯形图编辑。梯形图编辑是在梯形图编辑窗口中进行。可添加梯形图符号，删除梯形图符号，复制梯形图符号，剪切梯形图符号，粘贴梯形图符号，还可进行撤消、恢复、查找、替换等等。

输入的数据可为即时数，也可为所定义的符号。按要求确定。

编程一般按一个一个梯级进行。已有的梯级可合并，也可拆分。操作时在梯形图的相应位置点击鼠标右键，在弹出相应菜单后，再点击相应菜单项，即可实现合并或拆分。

编辑好的梯形图程序要进行编译。这按 CTRL + F7 键即可实现。编译的结果（程序的正确与否）会在输出窗口中显示。

图 2-107 所示为梯形图程序例子。

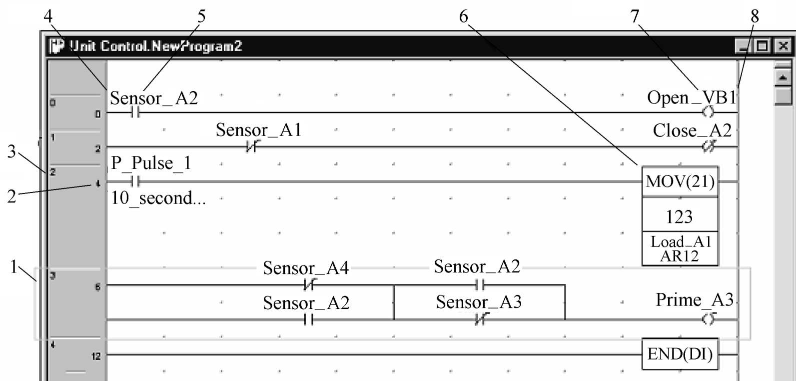


图 2-107　梯形图程序例子

1—梯级　2—步号　3—梯级号　4—左母线　5—接点　6—指令　7—线圈　8—右母线

程序编辑时，系统会对所编的程序进行检查。检查有 3 个等级（A、B 及 C）。等级不同，检查的项目也不同。检查的项目也可自定义。有关检查项目的选定，可在主菜单

项“PLC”中，点击“程序检查选项”，将弹出如图 2-108 所示的“程序检查选项”窗口。

弹出“程序检查选项”窗口后，可作相应选择，并点击“确定”按钮，此窗口即关闭。此后，系统在编辑时，即可按选定的项目进行所编程序正确性的检查。

（6）功能块编辑

1）系统功能块。如图 2-109 所示，在“文件”菜单项下有“功能块”项，再在其下有“从文件装载功能块”项。用鼠标点击该项，将弹出“选择 CX-Programmer 功能块文件”窗口，如图 2-110 所示。



图 2-108　程序检查选项选定



图 2-109　功能块菜单

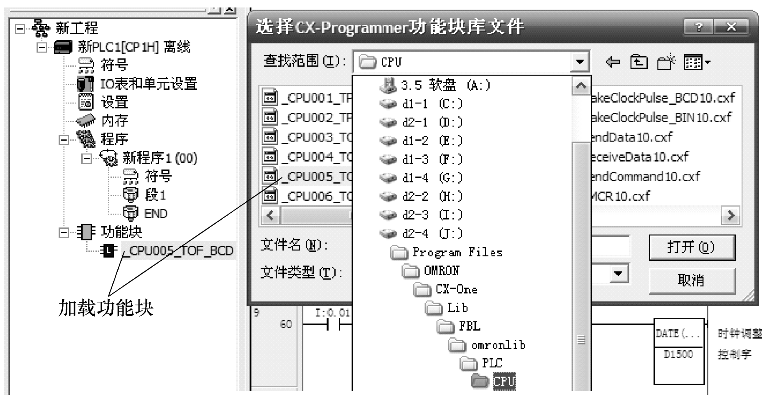


图 2-110　CX-Programmer 功能块文件窗口

该图显示“CPU”文件夹中的系统提供的库文件。如本例，用鼠标点击“打开”按钮后，选定“_CPU005_TOF_BCD10. cxf”文件，将装载用 BCD 码设置的 OFF 延时功能块。

OMRON 提供有与硬件单元对应的系统功能块。在安装 CX-one 软件时，会自动装载在“OMRON”的目录下的“Lib”项之下（如图 2-110 所示）。应用好这些功能块，可简化这些硬件的使用。

2）用户功能块。除了系统提供功能块，用户也可根据需要编辑功能块。为此，要先添加功能块。办法是，点击“插入”菜单下的“功能块”项。如图 2-111 所示，其下有 3 个

子菜单项。“梯形图”、“结构文本”以及“从文件”。点击“从文件”的效果与点击如图2-110所示的“从文件装载功能块”一样，为调用系统功能块。而“梯形图”、“结构文本”为自编功能块。前者用梯形图语言编写。后者用ST语言编写。

图2-112所示的功能块1、2即为新添加的自编功能块。打开它的属性窗口，可对功能块名称、是否显示内部以及作者、版本进行填写。同时，还可保护、注释作指定。

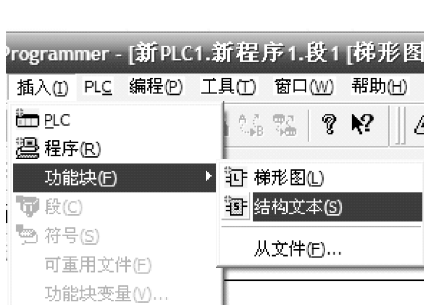


图2-111 插入功能块菜单

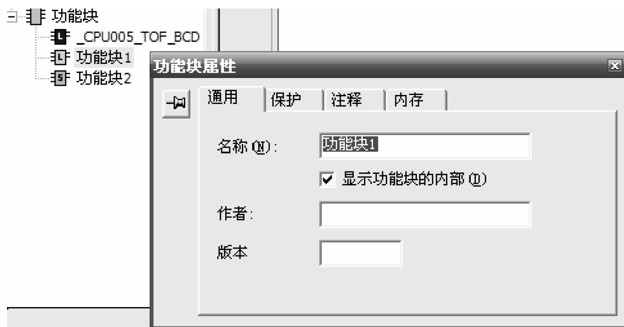


图2-112 功能块属性窗口

如果功能块使用梯形图语言编程，与主程序一样。所差的只是要声明输入、输出、内部变量。如果使用ST语言编程，对变量声明后，按要求输入ST语句就是了。

（7）程序注解。有多种注解，如：

1）变量注解。这在定义变量时进行。在梯形图显示时，将与变量名同时显示。

2）梯形图元素注解。这是对有关触点或指令所作的注解。做法是，选好要对其注解的元素，单击鼠标右键，将弹出一下拉菜单。选其中属性现单击之，则出现加注文本框。即可在其上写入有关注解。

3）标题注解。这是对工程、程序与段所作的注解。做法是，选好要对其注解的元素，单击鼠标右键，将弹出一下拉菜单。选其中属性现单击之，则出现加注文本框。即可在其上写入有关注解。

（8）查内存使用情况。在编程过程中，如要查看PLC内存使用的情况，可打开地址引用工具，或交叉引用报告窗口，从中可看到相应内存单元使用情况。

（9）其它工作。CX-Programmer有强大的编辑功能。可在所编的程序中，查找指令或变量。可复制、粘贴指令或数据。可替换数据。可删除指令或数据。

CX-Programmer还可作PLC的其它设定。这可在工作区窗口上，双击“设置”项后，将弹出PLC设置窗口。可在其上选定要设置的项目进行设定。设定后，要传送给PLC后才生效。

CX-Programmer还有文件存储，调用及打印功能。存储文件为单文件，其扩展名为“CX-Programmer”。存相同的文件名时，老文件可自动备份为扩展名为“BAK”的文件。

CX-Programmer可直接载入与处理以往OMRON公司其它软件编的程序（除LSS编的程序外）。只是有可能因地址或指令方面的差别，有的部分不能转换过来。为此，可作小量的人工修改或处理。之后即可升级，另存为CX-Programmer的文件。

至于用LSS软件编的程序，CX-Programmer还提供转换工具软件，“fileport. exe”。它封装在CX-Programmer软件包中，安装CX-Programmer后，自动也会装进这个软件。在Windows开始菜单的子目录中，查找“文件转换实用工具”项，并单击之，即可打开这个软件。

LSS 编程工具编的程序是存入程序包中。程序包的扩展名为“DAT”。转换工具软件可把这程序包中的一个个程序转换为 CX-Programmer 的文本文件，扩展名为“CXT”。经转换后的文件，CX-Programmer 直接即可处理。

2. 联机

联机是指计算机与 PLC 链接或联网，之间可传送程序或数据。

在建立联机操作前，一般要先对 PLC 与计算机的通信方法与通信参数作设定。PLC 与 PLC 联机最简单、最常用的是用计算机串口与 PLC 通信口。但，这个口通信的速度低。此外，可也可用控制网、以太网通信。通信速度快，但要有相应的联网硬件。

通信设定好后，计算机与 PLC 联好线，并把 PLC 接上电源，即完成了联机的准备。这时，点击在线工作菜单项，即弹出是否要联机的提示窗口，如回答是肯定，则将建立通信，计算机与 PLC 进入联机状态。

编程软件还提供自动联机的功能。有关通信参数可自动选择，为联机提供了方便。

提示：如联机失败，编程软件都将有相应提示。

在编程过程中，计算机与 PLC 联机主要要做的工作是：程序传送、远程操作及在线编辑。此外，还有数据传送，以实现对 PLC 的监控。

联机是指计算机与 PLC 链接或联网，之间可传送程序或数据。

（1）联机建立

在 PLC 设定时也要对 PLC 与计算机的通信方法与通信参数作设定。

一般用的是 Host Link 网，即 SYSMAC WAY 方式。选定后还要对驱动器作相应设定，所设定参数要与 PLC 的设定参数要一致。也可在 PLC 的 DIP 开关上选定通信参数为默认的。这样，计算机只要也按默认值选参数，即可实现联机、通信。

通信设定好后，计算机与 PLC 连好线并把 PLC 接上电源，即完成了联机的准备。这时，点击在线工作菜单项，即弹出是否要联机的提示窗口，如回答是肯定，则将建立通信，计算机与 PLC 进入联机状态。

CX-Programmer 软件还提供自动联机的功能。这可在“PLC”菜单项中，先单击“选择串口”项，选择要使用的串口。然后，再单击“自动在线”项，即可自动联机。在操作过程，CX-Programmer 都有提示，可提示进行操作。

（2）程序传送

进入联机状态后可向 PLC 传送程序（含 PLC 设定及有关数据等）。显然，如 PLC 未装有程序，未作必要的设定（或要改变默认的设定）则向 PLC 传送程序将是首先要做的工作。

下传的操作是：单击相应菜单项，或工具条，或操作相应热键。之后，将出现提示对话窗口，只要作相应的回答或选择，即进行下传。

如 PLC 中装有程序，或作了设定，也可将其上传给计算机。操作：也是单击相应菜单项，或工具条，或操作相应热键。之后，将出现提示对话窗口，只要作相应的回答或选择，即进行上传。

除了传送，还可把计算机的程序与存于 PLC 中的程序，设定及有关数据作比较。操作：也是击相应菜单项，或工具条，或操作相应热键，比较的结果也将有显示。

提示：新使用的三菱 Q 系列机，在下传程序设定及数据给 PLC 前，应先对 PLC 的内存进行格式化。格式化可用鼠标击“在线”、“格式化 PLC 的内存”菜单项，然后按提示操作。

（3）远程操作

远程操作是用以改变 PLC 的工作模式。具体操作可用菜单，或工具条，或热键进行。为确保系统安全，在进行这些操作时都有信息提示，并要求予以确认。

任何 PLC 都有两种基本状态：运行状态及非运行状态。处于前者时，PLC 运行程序，可实现程序的功能；但这时，多不能向 PLC 传送程序、修改数据，或对 PLC 进行设定。处于后者时，PLC 即使装载有程序，也不运行程序，不能实现程序的功能；但这时，可向 PLC 传送程序、修改数据，或对 PLC 进行设定。

在这两种基本状态中，不同的 PLC 多还有一些子状态，以便于用户对 PLC 的作不同的管理与使用。PLC 各状态间的切换，也各有各的办法，多不一样。

OMRON PLC 在运行状态中，还分有监控（在运行程序的同时，可修改数据，并可在线编辑，部分修改程序）及运行（不能修改数据）。它的大型 PLC 还有跟踪状态。初始模式设定。而且，这些状态都是由初始设定（确定加电后，PLC 所处于的状态）、简易编程器控制或用计算机远程操作。为了让计算机能向 PLC 写数据，控制 PLC，一般初始设定，都是使 PLC 一上电，都是处于监控状态。

提示：了解 PLC 的各个工作模式及其如何改变，也是使用 PLC 的一个基本要点。否则无法正确使用 PLC。

（4）在线编辑

程序下传后，如要作小量的改动，可进行在线（PLC 处运行状态）编辑。这时，PLC 仍运行程序、实现控制，同时，可接受所修改的部分程序。为了安全，在正式工作的场合，一般不主张在线编辑。但在程序调试时，在线修改，则是很方便的调试方法。

如程序是分模块的，也可按模块改，改后再下载修改过的模块，也是在线编辑。

CX-Programmer 可进入专门的在线编辑平台。办法是，先选好要改动的梯形图，再点击“程序设计/在线编辑/开始”菜单项，或热键，或工具条，则在梯形图所选定的梯级处即可进行与未联机前一样的梯图编辑了。编辑后，还要把编辑的结果传送给 PLC。这时，可点击“程序设计/在线编辑/发送修改”菜单项，或点击相应工具条，或相应热键，之后，CX-Programmer将对所作的改动作语法检查，如无误，则把所作的改动下传给 PLC。当然，如不想把所作的改动下传给 PLC，也可击“程序设计序/在线编辑/取消”菜单项，或点击相应工具条，或相应热键，之后，将退出在线编辑。程序也不会作任何改动。

3. 监控

与 PLC 联机还有一个目的就是与 PLC 交换数据，以对 PLC 进行监控。而且，也只有进行监控观察，才可看出所编的程序是否正确。每种编程软件都可在梯形图编程窗口上监控，还可在专门的显示内存数据的窗口上监控。有的还有其它监控方式。

与 PLC 联机还有一个目的就是对 PLC 进行监控。而且，也只有进行监控，才可看出所编的程序是否正确。本软件有多种方法进行监控。

（1）梯形图窗口监控。在联机后，点击“PLC/监视/监视”菜单项，或点击 CTRL + M，或点击工具条中切换 PLC 监视项，则进入或退出梯形图窗口监控。

这时，如 PLC 处运行或监控状态，则母线上有“电流”标志出现。触点通将有“电流”通过。等等。可形象地看到 PLC 的工作状况。

如显示的字体选择合适，还可在相应的指令显示处看到相应内存单元的当前值（即时

数据）。

图 2-113 所示为梯形图窗口监控：

在此窗口不仅可进行监视，还可写 PLC 的内存（在监控模式下）。可写通道（字），也可写（置）位（置为 1，或 0），还可强制置位。经强制后，此位的状态将不受程序或 I/O 刷新改变。

写或置位操作，可在梯形图中选好要写的内存地址（指令操作数），然后点击 PLC/强制（或置位，或设置）菜单项，或相应的热键，后再按提示进行操作即可。

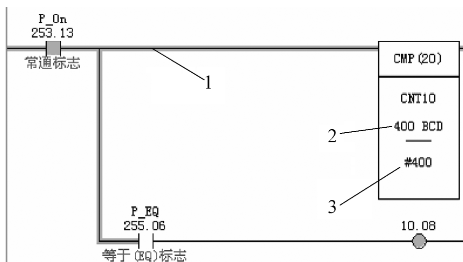


图 2-113　梯形图监控窗口
1—电流　2—即时数　3—设定值

不须强制可取消强制，办法也是点击 PLC/强制（或置位，或设置）菜单项，或相应的热键，后再进行强制取消的操作。

梯形图窗口对开关量，还可利用微分器进行微分监控。用此可观察到位的上升沿或下降沿的出现的情况，并伴随有声音及统计变化的次数。

选好要观察的位后，点击 PLC/微分监视器菜单项或相应的热键，即弹出如图 2-114 所示窗口。



图 2-114　微分监控设定

此图利用微分器监控的地址是“10.06”。

点击起始键后，开始监控，当 10.06 出现上升变化时，图 2-115a、b 两图将交替出现。



图 2-115　微分监控显示

这里的计数为该位出现上升沿的次数。

（2）观察窗口监控。首先，要打开观察窗口。然后，用鼠标指向观察窗口的对应处，双击之，等待弹编辑对话框。对话框出现后，在其上填入相应的地址。如不知地址名，可点击浏览键，点击后将弹出寻找符号窗口。可在其中找出要观察的符号地址。

图 2-116 显示的即为观察窗口及编辑对话框。

增加观察的地址后，如 CX-Programmer 监视状态，即可观察到该地址的现值。如 PLC 处于监控模式，也可在观察窗口写 PLC 内存。这时，先把鼠标指向要写的地址的列，并指向 PLC 名处，单击鼠标右键，等待弹出窗口，弹出窗口后点击数据设置，再在数据设置窗口写入要写的值。

（3）时序图监控。梯形图监控窗口激活时，点击“PLC/数据跟踪或时间图监视”菜单项。则弹出“PLC 时间图表监视器”窗口，如图 2-117 所示。

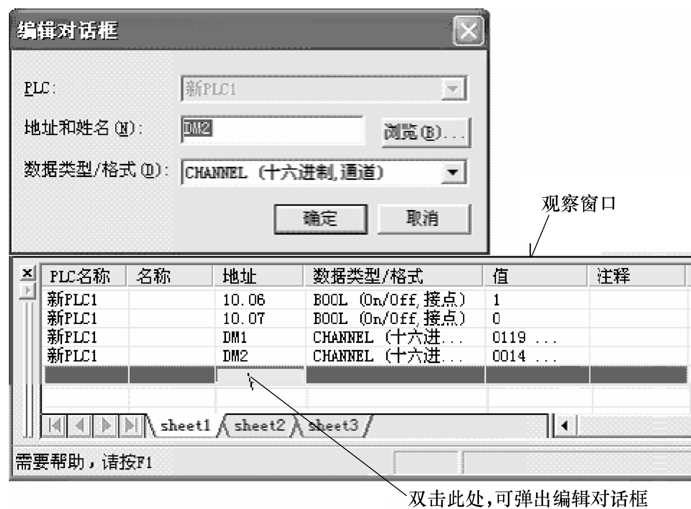


图 2-116　观察窗口及编辑对话框

在其上的“操作/模式”菜单项下，可选监控模式。但数据跟踪只在大型机才可进行。进行监控前首先要进行配置。

点击“操作/配置”菜单项将弹出配置窗口，如图 2-118 所示。

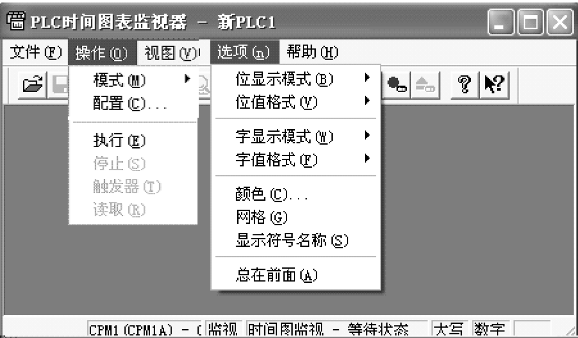


图 2-117　时间图表监视器窗口



图 2-118　配置窗口

这时可先选触发器项，由它选定触发信号及其特性。

选定后，点击“采样”，将改变为采样选择窗口，如图 2-119 所示。

这时可对采样时间间隔及其它参数作设定。

设定后，点击“字地址”或“位地址”，将改变为字或位地址窗口，如图 2-120 所示。



图 2-119　采样选择窗口



图 2-120　字地址窗口

这时可点击鼠标右键，将弹出如图 2-121 所示窗口。在其符号/地址项中可填入要监视的符号或地址。

如符号或地址不详，也可点击浏览键，将弹出，如在设置观察窗口时所看到的，寻找符号窗口。可在其上作相应的选择。



图 2-121　地址选定窗口

配置后还要点击工具条上的“执行跟踪/时间图”键，才能起动这个监控。起动这个监控后的画面如图 2-122 所示。

这里位与字显示的颜色与形式可在选项菜单中选定。监视后取得的数据可存为文件。

这种监控可从时序上看出各个量间的关系。所以对调 PLC 程序是很有帮助的。

（4）内存窗口监控。梯形图监控窗口激活时，点击“PLC/内存”菜单项。则弹出 PLC 内存窗口。此窗口与编程窗口类似，也可多文档工作，如图 2-123 所示。

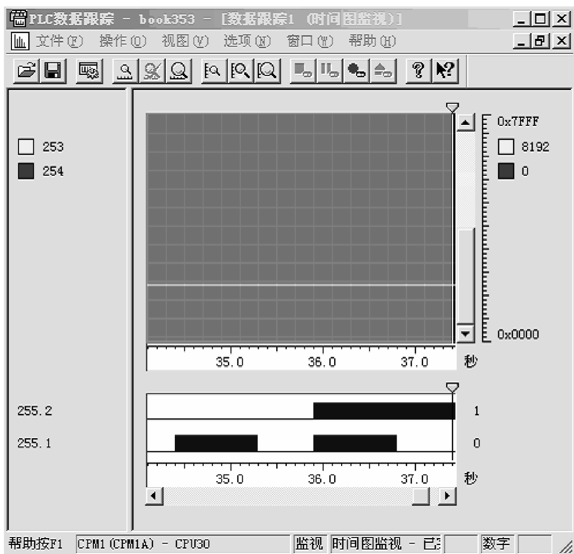


图 2-122　跟踪/时间图

本窗口可用于读（从 PLC 上传数据），写（向 PLC 下传数据，一般仅为 DM 区），及与 PLC 比较内存区。还可用于及时（采集时间间隔可设定）监视 PLC 内存数据。

监视时可任选内存区，也可自定相应地址。用后者时还可对位数值进行强制。图 2-124 所示为打开内存区进行监视 PLC DM 区 0050 到 0110 数据的画面。

如要监视其它 DM 区地址的现值，可用垂直滚动条进行调整。如要监视更多的 DM 地址，可增大窗口画面。如要监视其它内存区，可在画面的左区击 IR，SR，AR，HR 等，可按提示操作。



图 2-123　内存窗口

还可点击“地址”键，自定要监视的内存地址。图 2-125 所示为监视自定内存地址（255.0，255.2，255，10.0）的画面。

该画面仅监视四个量，还可增加。如图所示，在要增加处按鼠标右键，待弹出对话窗口，并

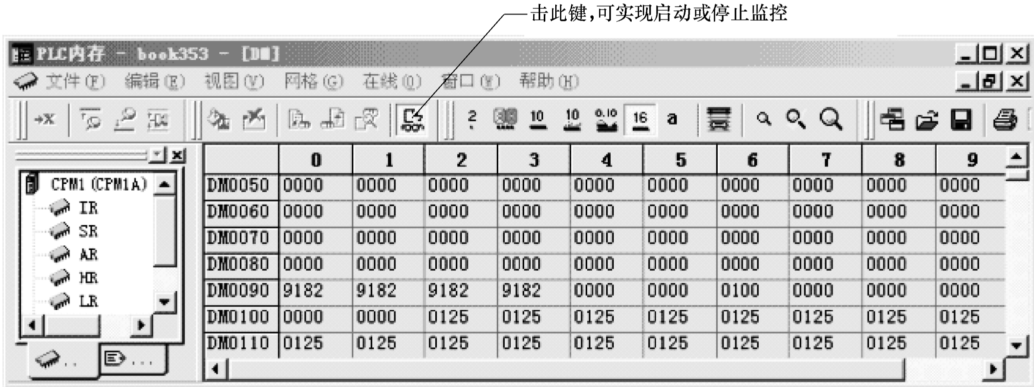


图 2-124　监视内存数据

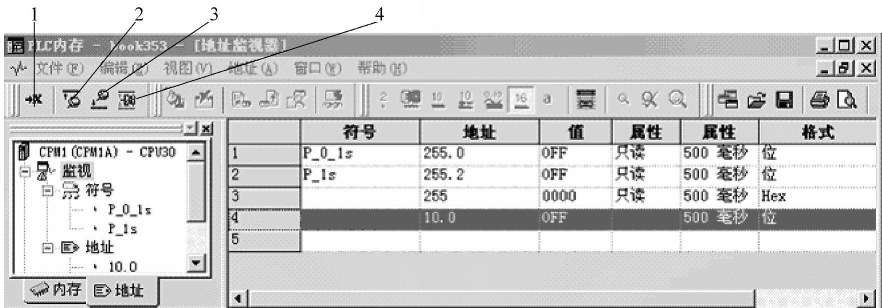


图 2-125　监视自定内存数据

1—设置　2—强制置位　3—强制复位　4—强制取消

在其中填入要监视的地址即可。

此窗口还可对 PLC 数据进行设置，强制置位，强制复位，或强制取消。选好要设置或强制的量后，击图中所示的工具条上的相应的键，则可实现。

图中属性 500 毫秒，为采集数据的时间间隔，可选。这可在显示数据区中，点击右键，即弹出如图 2-126 所示菜单，从中选定即可。



图 2-126　采集数据的时间间隔选定

所显示的数据的字体，大小，格式也可通过菜单，或点击工具条上相应的键选定，如图 2-127 所示，可选定显示格式。

此操作也可通过菜单实现，如图 2-128 所示。

选定内存区后，还可对其填充数据，如图 2-129 所示。填充后还可下载。只是多数 PLC 只能下载 DM 区数据。

除了下载数据，还可上载，或上下数据比较。等等。

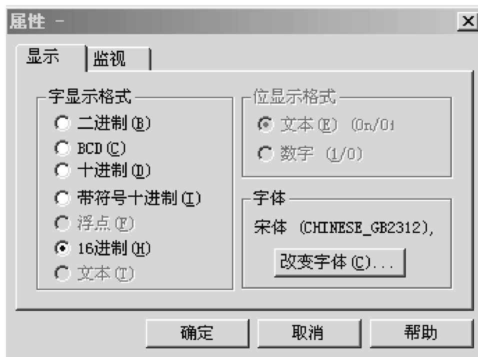


图 2-127　显示格式选定



图 2-128　用菜单实现选定



图 2-129　填充数据选定

2.6.5　帮助及其它

编程软件都有帮助系统。在这系统中，除了都有软件版本说明，还有软件使用及 PLC 指令及内部器件，特别是特殊继电器及存储器的说明。所以，这个帮助绝对是编程的重要支持。

CX-one 的帮助较详细。对指令的功能及使用都有具体。而且，进入帮助基本上是 Windows 风格，很方便。只要把焦点集中在指令上，按 F1 键，即可弹出帮助窗口。一个帮助不能尽意，还可再击“相关项目”，求得新的进一步帮助。

除了以上在线帮助，一般都有软件使用说明书，或电子文档，这些都可以从有关 PLC 厂家的网站上免费下载。

2.7　PLC 算法编程

关键词：算法、算法设计、算法表达、算法实现

算法编程是指，用优化的算法进行 PLC 程序设计。算法编程，要分析问题、寻求解决问题的算法；算法对比，选择最佳的算法，调用 PLC 指令及数据，去实现算法。

算法编程是在理论指导下的编程。既为编程入门提供指导，又为编程提高搭建台阶。是掌握 PLC 编程技术的科学之路。

2.7.1　算法概念

算法（Alogorithm）来自数学计算的概念。在公元前 200 多年，欧几里德的《几何原本》（Euclid′s Elements，第Ⅶ卷，命题 i 和 ii）中最早提到过它。从此，算法一词经常地同欧几里德算法联系在一起。

当年，为了求两个数的公因数，欧几里德提出辗转相除法，也就是后人称的欧几里德算法。如果以符号语言表达他的算法，将是：

```
WHILE z>0(如果 z>0,则重复下面的操作)
{
  z←x mod y;(x 被 y 整除,余数存于 z)
  IF z>0 (如果 z>0,则进行下面的操作)
  (
  x=y;
  y=z;
  )
  ELSE (如果 z=0,则公因数就是 x)
  公因数=x;(输出 x)
  }
```

当然解决这个问题还可以有别的算法。如，先把这两个数分解成若干质数相乘。再，在其中挑选出两个数共有的质数。这共有质数相乘即为所求最大公因数。

那什么是算法？从上可知，算法是指，解题的步骤。那什么是好算法？那就是步骤要少，每一步的含义明确、运算简单。运用它能够从已知数求得未知的结果。上述两个算法都有这些特征，而欧几里德算法则更好些。

更一般地讲，一个有效的、好的数学算法，一般有 5 个特征，即要有输入（已知数）、输出（未知数），同时还要满足有穷性、确切性、可行性。

PLC 主要用于控制，还可用于运算。当然，为了控制也要运算。所以，PLC 的运算是很多的。特别是逻辑运算更多。PLC 编程，就是编写好的控制与运算程序，让 PLC 一步步去执行，以实现所要求的控制与运算。

显然，这里的程序也有输入、输出，也要有穷性、确切性、可行性。具有算法的 5 大特征。

为了让 PLC 进行上述求两个正整数的最大公因数计算，如果用算法编程，那就要用程序去实现欧几里德算法。

在本书一开头，就讲到，“入出信息变换、可靠物理实现，可以说是 PLC 实现控制的两个基本要点”。所以，在本质上讲，PLC 只是输入到输出变换的工具。而输入到输出的变换是靠运行 PLC 程序实现的。所以，从广义上讲，PLC 程序也可说成，用 PLC 指令表达的 PLC 输入到输出变换的步骤。

简而言之，对照数学算法，PLC 的算法，则是用文字、公式、图形表达的、实现输入输出变换的、有限的、明确的及可实现的步骤。

可知，PLC 算法与 PLC 程序之间所差的只是表达方法的不同。算法可以说成是程序的原型、雏形、思路、提纲，而程序则是算法的成型、具体化及实现。

所以，一般讲，设计程序前，要先设计算法。这正与写文章一样，在写之前，一般要写提纲、打腹稿，或构想思路。

2.7.2　算法设计

算法设计就是寻找解决问题的步骤及各个步骤的工作内容。显然，问题不同，算法设计也将不同。以数学问题的算法设计为例，常用的有：穷举搜索法、递归法、回溯法、贪心法、分治法等。而为了使用这些算法，还可使用一些策略。如：

从输入推算到输出：演绎法；

从输出回求到输入：归纳法；

化整为零：规模化大为小；

动态规划：步骤从多到少；

模块化设计：从顶到底，由上至下。

等等。

PLC 编程算法也有数学问题，这些算法、策略也是可以借鉴的。但它更多的还是工程控制问题。所以，它的算法还与控制理论、信息处理理论相关。本书从第 3 章开始研究的就是，为解决各种工程控制及信息处理问题的算法设计。

算法设计是对问题的综合，不像问题的分析，可能有很多解决方案。所以，还有个算法优化问题。要选用最少的步骤、最简的操作，去达到最好的效果。

为此，需要对算法进行评价。评价算法主要依据是，算法的时间复杂度，空间复杂度。前者，主要看步骤多少及执行各步骤的时间。后者，主要看算法预计要使用的硬件资源。显然，所用的步骤越少，时间越短，使用的硬件资源越少，这个算法就越好。

本书以后各章的内容，主要也就是研究种种优化算法及其程序实现。

2.7.3　算法表达

算法表示的方法较多。凡能表达出算法思想、步骤，而又能让人理解的方法，都可使用。本节开始介绍的求自然数 xx，yy 公因数的欧几里德算法，就是用符号语言表达的。此外，还可用图解表达、图表表达及文字说明表达。

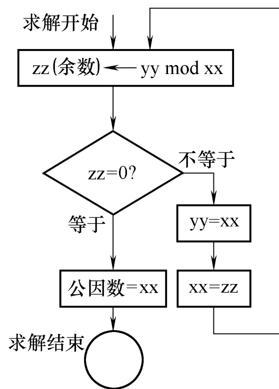


图 2-130　算法图解表达

1. 图解表达

图 2-130 所示的为 xx、yy 两个数的公因数的求解算法。图中方块表示“运算”，菱形表示“判断”，连线及其箭头表示“步骤及其联系”。

从图可知，这个表达是可清晰地反映上述求解过程的。

2. 图表表达

表 2-25 所示的为上述算法的求解图表。该表列了 5 个步骤。第 1 步为求解开始，进行求“余数计算”。第 2 步为“判断”，根据判断结果分成 2 个走向，或求解结束，或继续求解。显然这个图表也可清晰地反映上述求解过程。

3. 文字表达

上述算法也可用文字表达。可做如下表达：当求解开始后，先是求 yy 被 xx 整除，并把余数存于 zz 中。然后判断 zz 是否为 0？如等于 0，则 xx 即为公因数，运算结束；如不等于 0，则 xx 赋值给 yy，zz 赋值给 xx，并重复开始的步骤。

表 2-25 求解图表

步骤	内容		备注
1	zz(余数)←yy mod xx		求解开始
2	zz = 0?	等于 公因数 = xx	求解结束
		不等于 转步骤 3	
3	yy = xx		
4	xx = zz		
5	转步骤 1		

显然，文字表达是不够精炼的，特别是求解的问题较复杂时，更难以让人理解。故本书用的算法表达多是图解，或表格。个别也用一些解析式子。

2.7.4 算法实现

有了算法，还要编成 PLC 的程序，PLC 才能执行这个算法。这也就是算法的程序实现。它涉及问题有，指令选用及资源利用。

1. 指令选用

这与 PLC 编程语言的选择、编程软件或编程工具的使用及具体用什么厂家的 PLC，以及什么样的 PLC 密切相关。

至于具体选用什么指令，应做到：先用、多用高效率、执行时间短的指令。多用子程序、跳转、步进指令。以使所编的程序比较简练，运行时间短，程序的可读性好。

2. 资源利用

涉及到 I/O 分配及内部器件，如定时器、计数器、存储器的运用。I/O 分配除了考虑实现算法的方便，还要考虑硬件接线，及防止输入信号受干扰。

一般讲，PLC 的内部器件是足够多的，但也要运用得当，要有规律性，便于理解、便于查找、便于更换。

以下以求两个整数的公因数为例，看看，欧几里德算法是怎么实现的。图 2-131 所示的为上述求两个整数的公因数的子程序。

从图知，它就是实现欧几里德算法中，在花括弧内的算法。到了余数 z（DM7）为 0 时，即 200.02 ON 时，则输出公因数。否则，一直计算将继续。这里 xx，yy 为形式参数。调子程序前用 x，y 对其赋值。

提示：DIV 指令为整除运算指令。如本例，其商存于 DM6 中，而余数存于 DM7 中。故判断 DM7 是否为 0，即可知道 y 能否 x 整除。

图 2-132 所示为其主程序。

从图知，它就是实现欧几里德算法中，花括弧外的算法。0.00 为求解起动信号，一旦它 ON，将使 200.00 ON 一个扫描周期，并用它去起动 200.01，进而，调求两个整数的公因数子

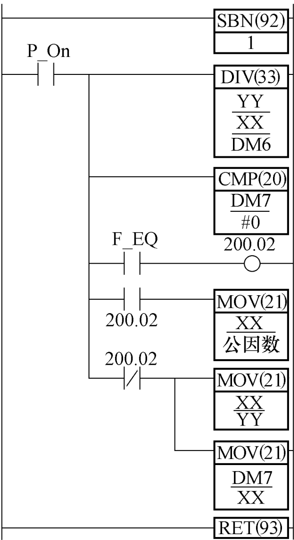


图 2-131 求两个整数的公因数子程序

程序。直到 200.02 ON（即 z = 0），并通过 200.03 使 200.01 OFF，不再调用子程序。

提示：图2-132程序用了微分指令及200.01的自保持功能，目的是确保在200.02 OFF，即 y 未能被 x 整除时，子程序一直在调用。而到了200.02 ON，即 y 能被 x 整除时，又可使200.01 OFF，子程序停止调用。在多周期完成运算的情况下，这么处理是很方便的。上述子程序也可改用功能块。ST 语言编写的 calculate 功能块，在编写前先定义变量。

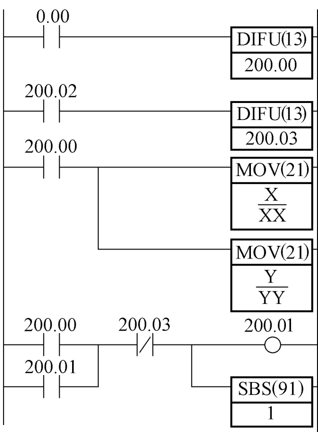


图 2-132　求两个整数的公因数主程序

输入变量：XX1、YY1，为 UINT（无符号整型量）

输出变量：XX、YY，为 UINT（无符号整型量）；AA BOOL（布尔量）

内部变量：YS、ZZ，为 UINT（无符号整型量）

功能块使用语句为

```
YS:=YY1/XX1;(*整除*)
ZZ:=YY1-YS*XX1;(*求余数*)
IF ZZ=0 THEN
    AA:=TRUE;(*如余数为0,计算结束*)
ELSE(*如余数不为0,通过输出XX、YY,重置XX1、YY1*)
    AA:=FALSE;(*如余数不为0,计算未结束*)
    YY:=XX1;(*变更被除数*)
    XX:=ZZ;(*变更除数*)
END_IF;
```

而图 2-133a 为调用它的部分主程序。主程序的其它部分与图 2-132 相同。而图 2-133b 显示的主程序，当 A ON 时，可一次调用功能块 calculate1。该功能块可连续计算，一次调用即可得出结果。其结果存放在 D20 中。

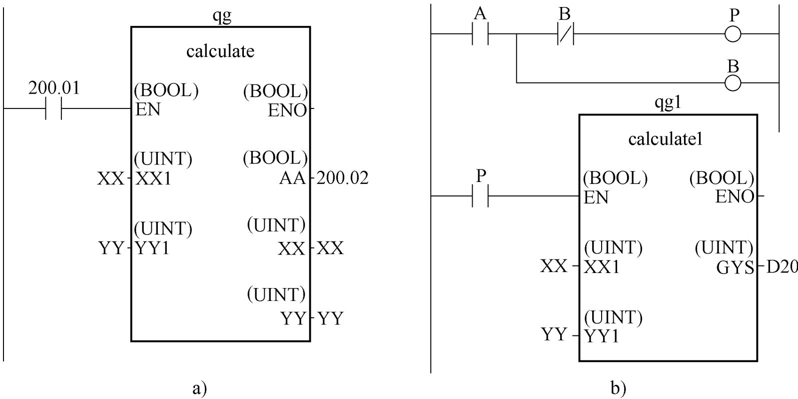


图 2-133　调用功能块主程序

功能块 calculate1 使用变量有：

输入变量：XX1、YY1，UINT（无符号整型量）；

输出变量：GYS UINT（公因数，无符号整型量）；

内部变量：YS、ZZ，XX，YY UINT（无符号整型量）。

功能块 calculate1 使用的语句为

```
XX：= XX1；
YY：= YY1；
REPEAT
  YS：= YY/XX；(＊整除＊)
  ZZ：= YY - YS＊XX；(＊求余数＊)
  IF ZZ > 0 THEN
  (＊如余数不为 0,重置 XX、YY＊)
  YY：= XX；
  XX：= ZZ；
END_IF；
UNTIL  ZZ = 0（＊如余数为 0,计算结束＊）
END_REPEAT；
  GYS：= XX；(＊输出公因数＊)
```

2.8 PLC 经验编程

关键词：经验、经验学习、经验积累、经验升华、经验应用

经验编程是指，用成功的经验进行 PLC 程序设计。也是很常用的编程方法。PLC 编程，从“不懂”到“懂”，从“懂”到“会”，从“会”到“熟”，最后做到“熟能生巧”，关键在于从实践积累经验。

如果说掌握理论有助于学会设计算法，那么编程技巧则主要来自经验了。没有实践，没有经验的积累，怎么能进入“熟能生巧”境地！

2.8.1 经验学习

经验有别人的，也有自己的，都很重要。前者要靠细心学习，后者要靠用心积累。都要花一定的时间与必要的精力。

别人的经验有上了书的或登载在杂志上的。本书介绍的例子程序，有的是我细心学别人的，但多数是我自己的经验。所有的例子都经我测试过，都经实践证明是可行的。我想，别的书本或杂志上介绍的经验也会是这样的。所以，学习这样成功的经验是必要的。

别人的经验还有没有上书的或没有登载在杂志上的，或许还是你同事的经验，也很值得学习。这种经验离你很“近”，很易借鉴。

自己的经验则是最重要的。要在自己的实践中，积累自己的经验。同时，最好在学别人的经验时，也能亲自作些测试，能使自己也有类似的经历，进而把这些经验变成自己的。这也是自己经验的重要的积累。

还有一些失败的经验，这往往是不会公开的，但这些经验也是要学习，也要积累。“失败是成功之母”嘛。

2.8.2　经验积累

经验的积累要用自己的脑记，更要用电脑记。最好作些分类，建立一个自用的程序“库”，以便于随时引用。

以下不妨列出，本人积累的一些，关于各种单按钮起、保、停逻辑的典型例子，供读者参考：

图 2-134 所示为用一次作用信号去起、停梯形图程序，曾在讨论“步进程序”时用过。这里不同的只是，一次作用信号是由“起、保、停”位通过微分（DIFU）指令产生的。图 2-134a 所示的是未工作的情况。这时，如加入一次作用信号，将使原 OFF 的工作位变为 ON、进入图 2-134b 状态。在图 2-134b 状态下，再加入一次作用信号，将使已 ON 的工作位变为 OFF、又回到图 2-134a 状态。

图 2-135 所示也是用一次作用信号去起、停一个工作位的梯形图程序，曾在介绍“单按钮起、保、停”程序讨论过。这里不同的也只是，一次作用信号是由起、保、停位通过微分（DIFU）指令产生的。它与图 2-134 所示不同的，只是它的控制位是先串后并，而图 2-135所示是先并后串。

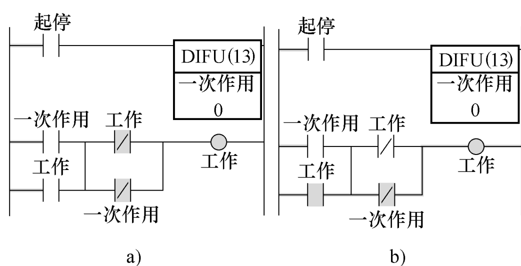


图 2-134　单按钮起、停工作过程

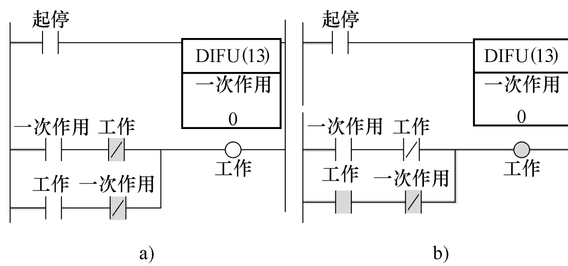


图 2-135　单按钮起、保、停逻辑另例

图 2-136 所示也是用一次作用信号去起、停工作一个位的梯形图程序。它用的是 KEEP 指令。图 2-136a 所示的是未工作的情况。这时，如加入一次作用信号，将使原 OFF 的工作位变为 ON、进入图 2-136b 状态。在图 2-136b，再加入一次作用信号，将使已 ON 的工作位变为 OFF、又回到图 2-136a状态。而一次作用信号则是也是由“起、保、停”位通过微分（DIFU）指令产生的。

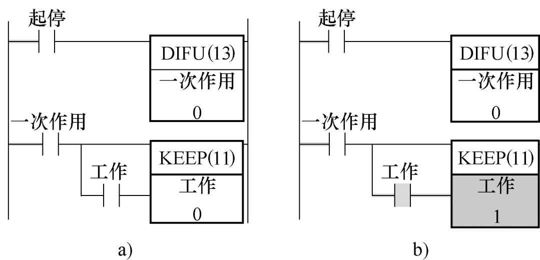


图 2-136　单按钮起、保、停逻辑又一例

图 2-137 所示也是用一次作用信号去起、停一个工作位的梯形图。它的起、停分别用 SET 与 RSET 指令。图 2-137a 所示的是 SET 指令在前执行的逻辑。图 2-137b 所示的是 RSET 指令在前执行的逻辑。效果是相同的。

读者也许注意到，在图 2-138 所示中。分别加有操作数为“10.09”位的上微分（DIFU，对图 2-138a）、下微分（DIFD，对图 2-138b）指令，且还用 10.09 位的常闭触点串入 RSET（对图 2-138a）、SET（对图 2-138b）的输入端。这么做是必不可少的。如无此，或指令的顺序作不适当的调整，都将无法实现这个功能的。

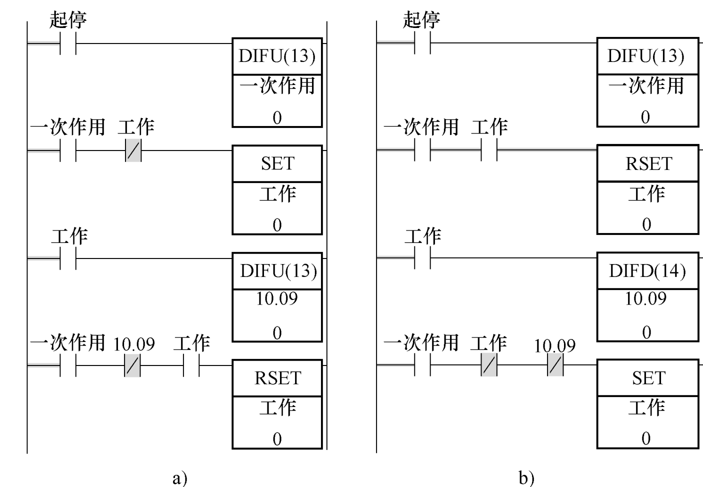


图 2-137 单按钮起、保、停逻辑再例

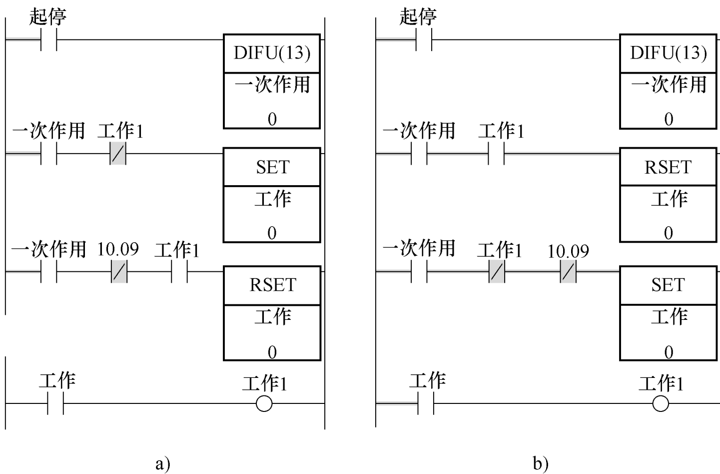


图 2-138 单按钮起、保、停逻辑再例续

图 2-138 所示是图 2-137 所示的续。它的起、停也是分别用 SET 与 RSET 指令。但添加中间变量“工作 1”。它在最后一个梯级时由“工作”赋值。未到执行这梯级指令时，即时“工作”状态变化，但“工作 1”不变化。这样，尽管图 2-138a、b 指令顺序不同，但效果是相同的。

此外，还可用计数器指令处理。也可实现上述功能。总之，只要用心积累，同一问题，多可归纳出多个解决方案。这不仅可拓宽自己算法设计的思路，还可更熟练地使用 PLC 的资源。

2.8.3 经验升华

经验还有待升华。升华有 3 个层次：

最低的层次如上述，可建立一些典型程序集（库），供今后再用。若程序较复杂，还可建一些功能块，或子程序，以便今后引用。

其次，要总结出有效算法。如以上介绍的单按钮起、保、停程序集，虽然实现的方法很多，但总结出它的算法不外是：

——执行一次指令

——如工作，转为停止

——如停止，转为工作

——但上述分支只能有一个选择。图 2-138，用了 10.09 位，就是保证实现这个算法的。

总共就只有以上四句话。凡按这四句话去设计的逻辑，肯定就能实现所要求的功能。用单按钮实现起、保、停，还有其它算法。这在本书第 3 章，还将讨论到。

最高层次的升华是，把经验上升到理论的高度，为丰富 PLC 编程理论作贡献。我想，随着 PLC 使用的普及与提高，是会有越来越多、从经验中升华出来的、而又能用以指导实践的 PLC 的编程理论的。

2.8.4　经验应用

经验积累、经验升华都是为应用。经验应用有三方面：

用作工程设计模板：设计新系统时，选用一个或几个与现有工程类似的、已取得成功的工程，作样板进行设计。这既可减轻设计的工作量，又增加设计的成功率。这也是信息可重用的一大好处。

用作编程参考：在无成功的工程可作样板时，在新设计的逻辑中，仍有相当一部分控制逻辑，可采用或借用已有典型逻辑，这也可减少设计的工作量，增加设计的成功率。

用作算法设计参考：在既无样板可参照，又无典型可采用时，还可运用过去的一些成功的算法。

经验是宝贵的，但是，经验，特别是个人经验，总是有限的。所以，经验的应用也还要与编程理论相结合。而本书所作的就是，结合个人积累的编程经验，作些编程理论研究，为与读者经验的结合提供参考。

结语

“工欲善其事，必先利其器!”PLC 编程这个“事”的“器”就是编程软件。要编好程序，当然必须有好的编程软件，这是 PLC 厂家要做的。对 PLC 用户而言，就是要弄清怎么用好这个软件。一定要熟悉它的组成、功能以及如何发挥使用好它的功能。最少要大体了解本章对其所作的简要介绍。这是其一。

其次要弄清 PLC 的资源。即 PLC 的编程语言、指令系统及相关软器件的情况。都使用了哪些语言？PLC 都有哪些指令或语句？各能实现什么功能？如何使用它的功能？都有哪些软器件？各能实现什么功能？如何编址及使用它的功能？等等。本章详细介绍图形图及 ST 语言，指令系统及语句，软器件及其应用，目的也是为此。

显然，弄清以上问题还不是目的，真正的目的是编程。为此，还要弄清解决问题的步骤，即算法。把算法用指令或语句及数据去表达，即为程序。显然没有解决问题算法，也就没有解决问题的程序。而算法则与解决的问题性质有关，这将是本书要逐步研究的课题。所

以，仅仅学习本章还是不会编程是不奇怪的。但为了熟悉程序是怎么回事，本章特提供的若干典型程序及相关说明，可作为了解程序的参考。

本章的目的只是为编程建立基础。而且，也只是建立初步基础。想都弄清它之后再去编程也不现实。进一步建立基础，也可能要这样做：先大体了解本章介绍的编程软件及 PLC 资源，再在往后的章节介绍中，根据要解决的问题研究算法，最后再在算法实例化中进一步熟悉编程软件及 PLC 资源。这些过程反复循环，不断提高，将使你的编程进入新的境界。

请想想

1. 指令的本质是什么？编程语言的本质是什么？梯形图编程语言及 ST 的特点是什么？
2. 软器件的实质是什么？怎样才能使用好它？
3. PLC 数据内存器件化的利弊？发展趋势？
4. 堆栈、结果寄存器的作用？
5. 何谓中断屏蔽与禁止？
6. 用步进处理的顺序控制优点是什么？
7. 对指令理解有那几个要点？
8. 用符（标）号地址编程的好处？
9. 算法是什么？为什么要研究算法？
10. 编程软件都有哪些基本功能？

请试试

1. 熟悉编程软件使用。
2. 用编程软件录入典型程序，下载给 PLC 并进行调试，观察是否可实现所要求的功能。
3. 设计以下程序：

定时通断程序、用计数器计时程序、求最小公倍数程序。

4. 解锁程序。

第 3 章　PLC 顺序控制程序设计

逻辑量，也称开关量，是指，仅有两个取值，0 或 1、ON 或 OFF、TRUE 或 FALSE 的量。控制是指，使系统的状态与行为产生所希望的变化而加给系统的作用。如果表达这个作用及系统状态用的主要是逻辑量，则这个控制即为逻辑量控制。它是系统工作最常用的控制。

逻辑量控制的目的是，根据当前与历史输入的组合，使 PLC 产生相应的输出，以使系统能按一定顺序工作。所以，也称为顺序控制。

顺序控制编程是逻辑问题的综合，又是工程控制技术的集成。同样的问题可以有多个解决方案。设计者的任务就是，能在众多的方案中，求得最佳方案。

本章将讨论顺序控制的有关理论、算法设计及其程序实现，并介绍有关设计实例。

3.1　顺序控制类型及编程方法

关键词：控制、手动控制、自动控制、组合逻辑、异步时序逻辑、同步时序逻辑、确定电路、随机电路、集中控制、分散控制、混合控制

3.1.1　顺序控制类型

可从不同角度进行顺序控制分类：

1. 按人工干预情况分

（1）手动控制

用主令器件，如按钮，直接向系统发送命令以实现控制。

（2）半自动控制

一旦系统人工起动、系统工作，其过程的展开是自动实现的，无须人工干预。但在过程结束时，系统将自动停车。若再使系统工作，还须人工起动。

（3）自动控制

一旦系统人工起动、系统工作，其过程的展开是自动实现的，无须人工干预。而且可以周而复始地循环进行。若要使系统停止工作，则需要人工另送入停车信号，或运用预设的停车信号。

一个顺序控制系统，往往都具有这 3 种控制。有时系统较复杂，可能无自动控制，或者也无半自动控制，但手动控制总是有的。作为一种目标，总是要力求能对系统进行半自动，以致自动控制。

2. 按顺序的确定性分

（1）确定电路

控制对象工作过程或顺序是确定的。是用得最多的顺序控制。

（2）随机电路

控制对象工作过程或顺序不是确定的。也可理解为工作顺序是有分支的。它可依输入逻辑条件的不同，选择不同进程分支。

组合电路多为随机电路。如设备的各种手动操作，多是随机的。但时序电路也有随机的。如电梯，某一时刻往第几楼开，在第几楼停，要依使用电梯的人的要求而定。

图 3-1b 为一个简单的随机梯形图程序。它用以控制液压搬动手柄。其输入要求用 3 个按钮（A1、A2、A3，对应的输入点为 0.01、0.02、0.03）。体现当前手柄位置的用 3 个行程开关 XK1、XK2、XK3（如图 3-1a 所示）反映，并设其对应的输入点为 1.01、1.02、1.03。控制要求是，不管手柄处在何位置，按下 A1 后，应使其转到使 XK1 合下的位置；按 A2，则到使 XK2 合下的位置；按 A3，则到使 XK3 合下的位置。至于 A1、A2、A3 按哪个？什么时候按？是随机的。

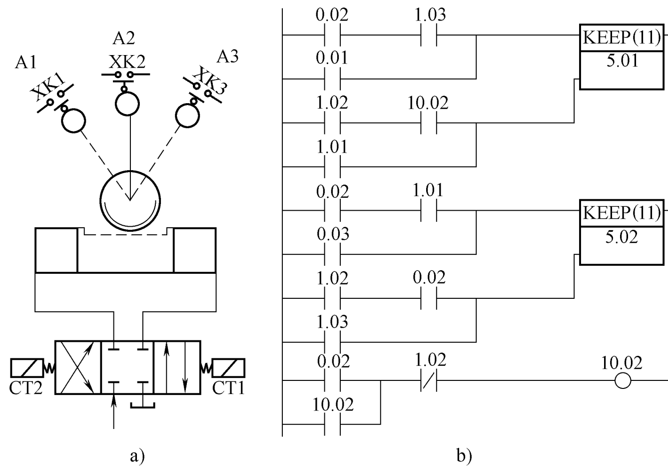


图 3-1　简单随机梯形图程序

手柄转动由液压机构操纵，如图 3-1a 所示，当 CT1 得电时，三位四通阀左移，使操纵手柄的油缸左腔与压力油通，右腔与回油箱通，柱塞在压力油的作用下向右运动，通过齿轮齿条机构，带动手柄逆时针转动。当 CT2 得电时，情况相反，手柄顺时针转动。CT1、CT2 都失电，阀处于中间位置，切断油路，手柄不转。

3. 按逻辑问题的性质分

（1）组合逻辑

输出仅与输入的现况有关，而与输入的历史情况无关。

如图 3-2a 所示，它是三个位置刀架示意图。图示这个位置，两个行程开关压合上。顺时针转 120°，则 XK2 压上，XK1 不压。再顺转 120°，则 XK1 压上，而 XK2 不压。再转 120°又回到图示位置。这样，用两个开关的被压合状况的组合，就可反映出刀架处于 3 个位置中的哪一个。

图 3-2b 是反映这个指示状况的梯形图。这里 WW1、WW2、WW3 哪个 ON（如用其驱动指示灯）？就只取决于刀架的当前位置，而与其历史情况无关。

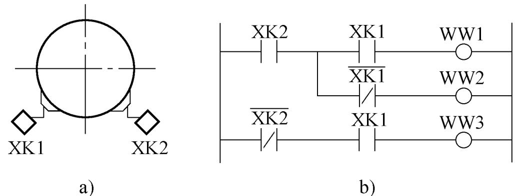


图 3-2　组合逻辑电路

从上述的组合电路知，它的特点是：

① 无反馈。其输出线圈仅受输入信号控制。

② 单一解。输出的状态仅由输入元件现状的组合直接反映，其结果是惟一的。

③ 电路状态转换，可以一次实现，没有中间状态的过渡。

④ 所需输入元件多，但电路简单可靠。当然，这么简单的电路是无需使用 PLC 的。这里只是分析方便，举它作为例子而已。

（2）时序逻辑

输出不仅与输入的现况有关，而且还与输入的历史状况有关。

图 3-3 是由图 3-2 改成的异步时序逻辑。它只用一个行程开关，其梯形图程序如图 3-3b 所示。

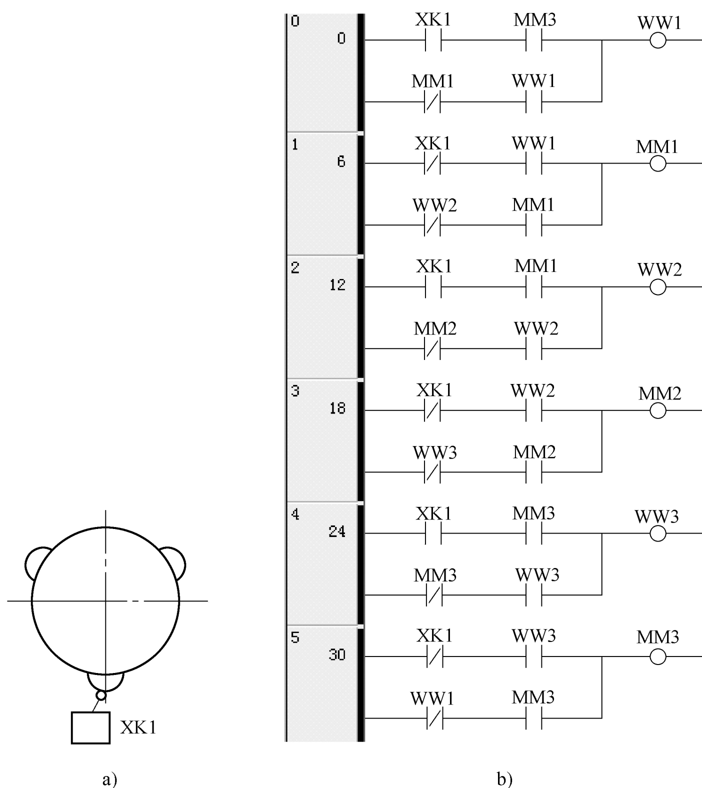


图 3-3　异步时序逻辑

a）刀架示意图　b）异步时序位置显示程序

如图 3-2a 所示，开始时，如 XK1 压上、ON，WW1 ON，显示处于位置 1。而当刀架逆时针转动，XK1 先松开、OFF，后到位又压上、ON 时，则 WW2 ON，WW1 OFF，显示处于位置 2。而当刀架又逆时针转动，XK1 又先松开、OFF，后到位又压上、ON 时，则 WW3 ON，显示处于位置 3。这时，刀架再逆时针转，则回到原来位置，WW1 ON，WW3 OFF。逻辑复原。

以上分析比较粗略，在下一节进行通电表介绍之后，还有详细说明。

当然，这种电路，刀架只能固定在一个转动方向。而且，这个逻辑在开始时，要根据刀

架的位置指定 WW1、WW2 或 WW3 哪个 ON。在实际上，刀架也多是固定一个方向转动的。故这个指定是不难实现的。

分析以上逻辑可知，时序逻辑的特点是：

1）多解。同样输入组合产生不同输出。它除了产生等效输出，还可产生长效、短效输出。

2）反馈。其输出线圈不仅受输入信号控制，还受（直接，或间接的）自身触点的控制。

3）顺序。多解的实际取值是由其工作顺序确定的。工作顺序是一步一步，或一个节拍一个节拍展开的。而步、节拍的转换总是由输入引起的。

4）所需输入元件少。它有记忆功能，可用以节省输入元件。这对于一些难以使用较多的输入信号的场合，是一个很大的方便。但在逻辑的关系上，时序逻辑相对也比组合逻辑复杂些。

时序逻辑电路又有同步时序与异步时序之分：

1）同步时序：有统一的节拍转换脉冲，输入信号改变的作用由节拍脉冲激活予以实现。这可避免在变量变化时，其间相互的影响，因而考虑的关系要简单一些。

2）异步时序：无统一的节拍转换脉冲，节拍转换由输入信号的改变实现。但变量的变化会相互影响，考虑的关系要复杂一些。

PLC 的 CPU 硬件逻辑电路就是同步时序逻辑。它的工作就是由同步（时钟）脉冲统一控制的。但 PLC 程序，只能是一条一条执行。从这个意义上讲，它的逻辑关系不是同步的，而是异步的。但有两点值得注意：

1）PLC 程序是循环执行，执行后还要靠 I/O 刷新予以实现。从这个意义上讲，是否也是同步？可否做到指令的执行是异步的，指令的实际作用是同步的？

2）PLC 有微分指令：上升沿微分，只在被微分量从 0 到 1 的那个扫描周期，这个微分输出才为 1；下降沿微分，只在被微分量从 1 到 0 的那个扫描周期，才为 1。可否用这个微分信号做同步脉冲信号，也按同步的方法设计部分梯形图逻辑？

答案是肯定的。因而也完全可以按同步逻辑的设计方法，设计顺序控制程序。这样处理，称异步时序逻辑同步化。

图 3-4 所示的为同步时序逻辑，它已作了同步化处理。

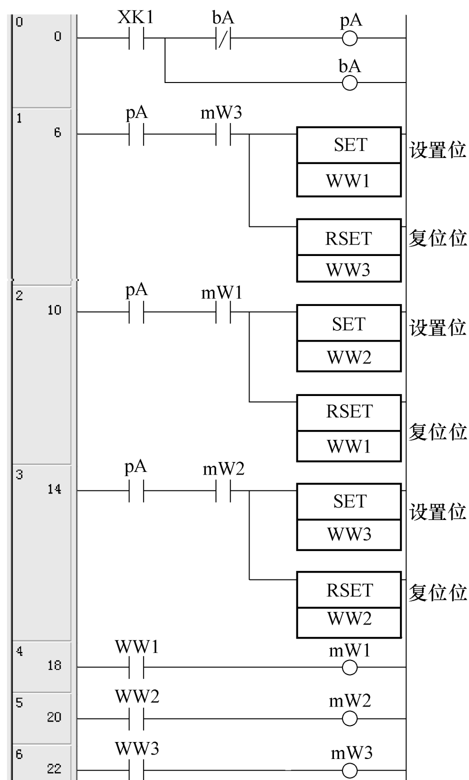


图 3-4 刀架位置显示同步时序逻辑

如图 3-4 所示，梯级 1、2 为把 XK1 信号转换为脉冲信号 pA。梯级 3、4、5 为程序主体。在执行中，它的输入条件除了脉冲信号，其它的都

不变。梯级 6、7、8 为状态记录。把输出变化传递给内部状态变量，即 WW1、WW2、WW3 分别传递给 mW1、mW2、mW3，为新的脉冲输入建立新的条件。仔细分析此逻辑可知，它也是用 WW1、WW2 及 WW3 显示刀架位置。

提示：时序逻辑同步化的两个基本点：一是只有脉冲信号能改变内部状态或输出；二是在脉冲信号改变内部状态或输出的期间，其逻辑条件不变。

3.1.2　顺序控制编程方法

逻辑量控制的编程方法很多，实用的有：基本逻辑设计法，图解设计方法、高级逻辑设计法、工程设计法及数据结构设计法。

此外，还有什么方法也不讲，只凭经验设计。有的继电电路设计的工程师，也是可先设计继电控制电路，然后再翻译成 PLC 程序。显然，这都不是好方法。

1. 基本逻辑设计法

基本逻辑设计法是基于逻辑量间的与、或、非的关系，用逻辑综合方法，处理逻辑量的输入与输出间的关系。

顺序控制程序逻辑设计就是，根据要求的逻辑量的输入（历史值与当前值）与输出间的对应关系，运用逻辑综合方法，运用 PLC 的基本逻辑处理指令去设计程序。

逻辑综合涉及到算法设计、PLC 的指令选择及资源利用。具体的答案往往不只一个，而是多个。但由于逻辑设计的理论比较成熟，所以，实现算法设计及编程优化是不难做到的。

从本质上讲，基本逻辑设计法与数字电路的综合是一致的。所以，如果有了数字电路的基础知识，对其进行设计是不难的。

基本逻辑设计法编程比较严谨，所设计的程序比较精炼。但若处理的变量多时，不大好把握。同时，基本逻辑设计法主要是基于与、或、非等逻辑关系的指令，而实际上，PLC 还有很多其它可供逻辑量处理的指令，所以，如果处理的问题较复杂，还应寻找较易掌握及较高效率的方法，这个方法也就是高级逻辑设计法及工程设计法。

本章介绍的基本逻辑设计方法有 3 个：组合逻辑设计法、异步时序逻辑设计法及同步时序逻辑设计法。

2. 图解设计法

它是运用图形进行设计。用梯形图语言编程实质也就是图形法，无论什么方法，若把 PLC 程序等价成梯形图后，就要用到梯形图法。

此外，还有时序图法、流程图法等。

时序图法是根据信号的时序关系，画时序图，再根据这个时序图，去分析信号间的关系，进而去设计程序。时序图法很适合于时间顺序关系清晰的顺序控制程序设计。

流程图是用框图表示 PLC 输出与输入之间的关系，在使用步进指令的情况下，用它进行设计，是很方便的。

图解法比较直观，设计过程不易出错，是人们较爱用的方法。

3. 高级逻辑设计法

高级逻辑设计法也是用逻辑综合方法处理逻辑量的输入与输出之间的关系。但与基本逻辑设计法不同的是，在逻辑分析时，它不单纯基于逻辑量间的与、或、非的逻辑关系，还要用到其它的逻辑量间的关系。在逻辑综合时，所用也不仅仅是 PLC 的基本逻辑处理指令，

还要使用到 PLC 的高级处理指令。

使用这些方法，既可把一些复杂的逻辑问题的分析变得简单，而且，还可充分利用 PLC 的资源，设计出效率较高的 PLC 程序。

4. 工程设计法

工程设计法就是运用自动控制理论与方法，分析与设计逻辑量输入与输出间的顺序控制关系。其目的是弄清所控制的对象，是怎样按要求的顺序工作的。

在 PLC 出现之前，工程上已用了很多顺序控制的方法，也有很多好的机理与做法。如普通金属切削机床的分散控制，就是靠每一动作完成的反馈信号，去结束本动作，并起动下一动作。到了所有动作都执行完了，如半自动机床则停车；如自动机床则靠最后一个动作的完成信号，去重新起动第一个动作。

对于生产率很高的自动机床，其自动化用的则是集中控制。手段是用分配轴、用机械的办法。机床工作时，分配轴不停地转动，并靠轴上的各种不同形状的凸轮，去控制机床的各部件运动。分配轴转动一周，各部件的运动也完成一个周期，并完成一个工件的加工。机理不复杂，但能完成很复杂的控制。只是它不是反馈控制，有点不可靠（只好用安全销作安全保护）。由于使用机械的手段实现控制，调整也较麻烦。

现代化的、功能很强、性能很高的机床多用混合控制。手段不用机械，而用电、用程序控制。机床按一个个程序依次工作。各个程序执行什么动作，可预先设计。而程序的转换则要靠动作完成的反馈信号控制。得不到反馈信号，程序将不会往下进行。所以，它既能完成复杂的控制，又能保证安全工作。

如果抛开上述控制的具体过程，把这些机理上升为控制算法，并选择 PLC 的有关指令去实现这些算法，就有了分散控制、集中控制及混合控制算法及相应的控制程序。只是以前金属切削机床自动化用的是硬件实现，而如今 PLC 控制机床工作用的是软件实现。

（1）分散控制。其控制命令是由分散的动作完成法反馈信号提供。图 3-5 所示的是基本分散控制的原理图。

从图中可知，当起动条件满足，给系统起动信号后，系统将产生动作 1。当动作 1 完成，则产生动作 1 完成信号，并用此信号直接使动作 2 工作（动作 1 停止）。到了动作 2 完成，再用它的完成信号使动作 3 工作……依此类推，直到所有动作完成，系统工作停止，或又从动作 1 开始，重复这个过程。

分散控制有反馈，工作可靠。其缺点是控制关系复杂，程序量随着动作的增加而增大。

（2）集中控制。命令是由集中控制器提供。图 3-6 所示的为按集中原则实现控制的简图。

这里控制器顺序地发出一个个动作命令，直到全部动作完成。如果是自动工作，控制器又发出动作 1 命令；如果不是自动工作，控制器停止发命令，系统不再工作。

用这一原则进行控制，其程序容易设计，效率高。有的 PLC 有

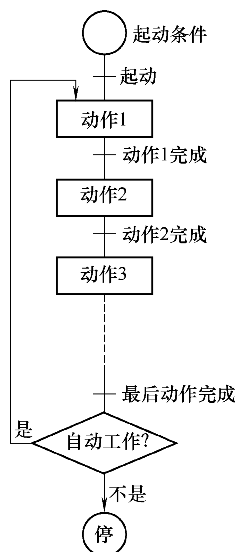


图 3-5 基本分散控制原理图

凸轮单元或凸轮指令，更为用这种原则的设计提供了方便。

其缺点是没有反馈，如果协调不好，或采取的措施不当，系统易出现问题。

（3）混合控制。它的控制命令由集中控制器发出，而什么时候发出命令，则是由分散的反馈信号控制。取决于反馈的条件满足与否。图 3-7 所示的是基本混合控制逻辑的原理图。

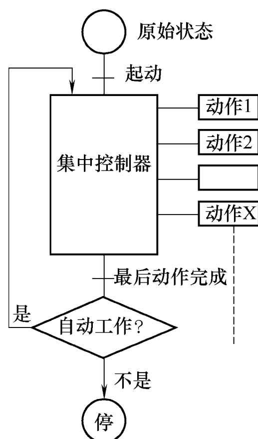


图 3-6　集中控制原理图

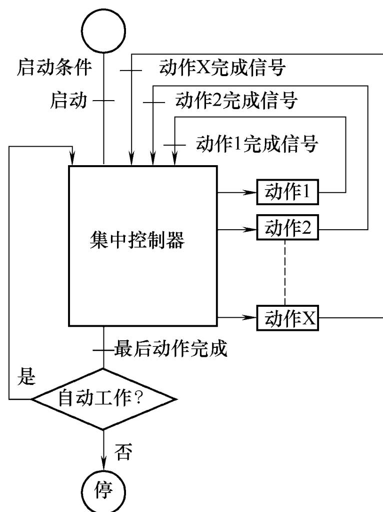


图 3-7　混合控制逻辑原理图

混合原则把分散与集中原则的优点兼而有之。程序要复杂一些，但不随动作的增加而增加。多用于复杂的顺序控制。

此外，计算机数据结构中用的线性链表的数据查找，与混合步进控制颇为类似。图 3-8 所示为线性链表算法的框图。

从图中可知，起动之后，启用链表中标号 1 的数据。用它的指向，生成设定输出，再由设定输出产生虚拟输出，进而通过逻辑转换变为实际输出，以进行对系统第一步控制。

之后等待控制效果的反馈。过程先把实际输入转换为计算输入，再进行计算输入与这里的设定输入比较。当这个比较结果一致，则转到次一标号，停用本标号指向的数据，启用次一标号指向的数据。接着，又等待新的控制效果的反馈。

这个过程重复进行，直到次一标号为 0，则过程结束，完成整个的顺序控制。

可知，它与混合控制不同的只是，它不用步进机制，而直接用数表链接。算法更简洁些。熟悉计算机编程的人员也许更愿意使用。只是，使用此法要求 PLC 除了能定义数组，最好还能定义结构。OMRON PLC 目前还不能定

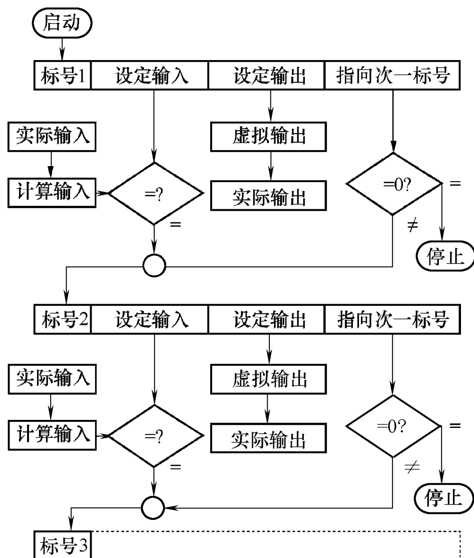


图 3-8　线性链表算法的框图

义结构。而国产和利时 PLC，LM、LK 型机，则可以。这个算法的程序实现，请参阅作者的专著《PLC 系统配置及软件编程》第 7 章。

3.1.3 顺序控制输入器件

1. 主令器件

用于产生控制命令的元器件，如有按钮、钮子开关及转换开关等。

（1）按钮。有常开触点及常闭触点。如未按下，则常开触点断，而常闭触点通。如按下，则常开触点通，而常闭触点断。而使用的通断手段可以为触点，也可为半导体电路。其工作示意图及电气图形符号如图 3-9 所示。PLC 使用的都是常开触点。

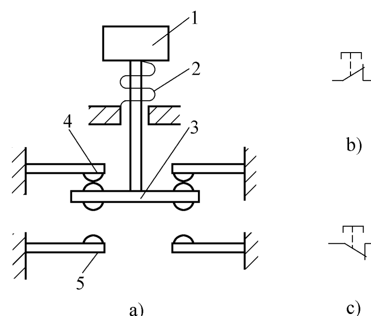


图 3-9 按钮

a）示意图 b）常开触点 c）常闭触点

1—按钮帽 2—复位弹簧 3—动触桥

4—常闭触点 5—常开触点

提示：PLC 用按钮多只使用常开触点，故可选用不带常闭触点的按钮。但也有人建议，为安全计，安全控制用按钮可以采用常闭触点。

（2）钮子开关。用于通过小电流。有“刀”与“掷”之说。前者指多少通路，后者指多少位置选择。如两路、两掷，指可控制两个通路，有两个接通可能位置选择。即可使其通，还可使其断。再如两路、三掷，指可控制两个通路，但每个都有三个可能位置选择。家用照明控制开关多是这类开关。

（3）转换开关。通过电流较大。而且，“刀”与“掷”更多，可用于控制命令的多种选择。

2. 反馈器件

用于生成触点通或断反馈信号的元器件。

根据生成的原因分，有：行程开关、压力继电器、速度继电器、热继电器、液位继电器、时间继电器、计数器、电流继电器。

（1）行程开关。是检测部件运动行程反馈器件。有常开触点及常闭触点。如未达到行程要求，则常开触点断，而常闭触点通。如达到行程要求，则常开触点通，而常闭触点断。而使用的通断手段可以为触点，也可为半导体电路。后者也称接近开关。

（2）压力继电器。是检测气体或液体压力信号的器件。达到设定或超过设定压力时，它常开触点通，而常闭触点断。反之，则常开触点断，而常闭触点通。

而使触点通断手段可以是实际的通断的触点。也可为半导体电路。前者称触点器件，后者称无触点器件。无触点器件用改变电路电阻实现通断。加大电阻，相当于断开；减少电阻，相当于接通。

（3）其它物理量继电器。如速度、温度等，也都可成为检测量，也有对应的继电器。而由于 PLC 内置有大量的“软”计数器及时间继电器。所以，这两种继电器在 PLC 控制系统中基本不用。

3. 输入接线

（1）触点输出的传感器件接线。多只用常开触点。用它的常闭触点则对它取“非”。如使用停车按钮，就是这么处理的。

触点是无源的，没有极性的区别。所以，与 PLC 输入点连接时要把电源串入。如图 3-10a所示，电源可以使直流的，如 24V，也可交流的，如 110V、220V。器件距离 PLC 较近，多用24V 直流。因为，PLC 多提供有此电源，PLC 输入模块的输入点电压也多是这个规格，所以，这么用很方便。当然，如果器件离 PLC 较远，为了防干扰，也可用交流电源。但这时一定要选用相应电压规格的输入模块。

（2）无触点输出的传感器件接线。这类传感器是有电源的电子电路，有极性的。针对不同的输入传感器有不同的接线，如图 3-10 所示。由于电子电路分断时总是有小量的漏电流及上电时出现浪涌电流，很可能出现信号被 PLC 误读。所以，有时要采取措施予以避免。具体细节请参见 OMRON PLC 操作手册。

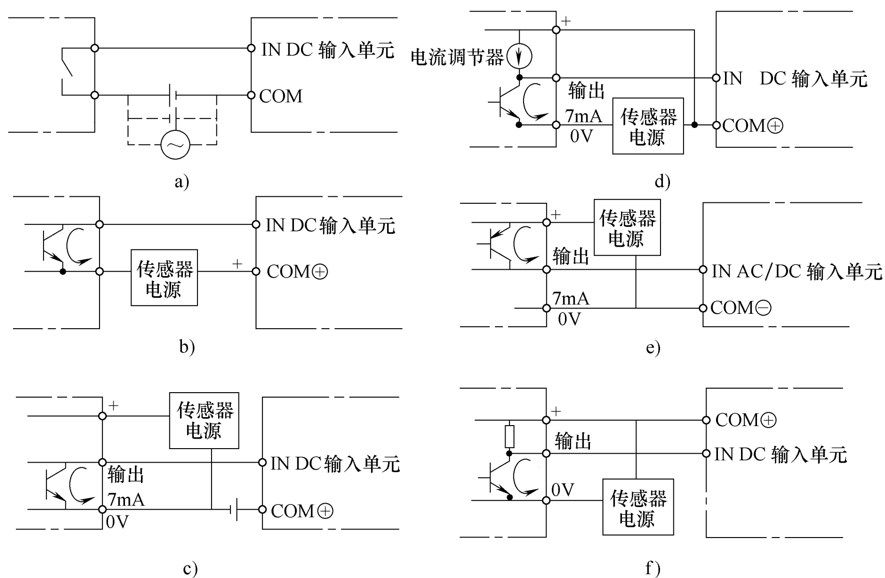


图 3-10　不同输出的传感器与 PLC 输入点接线

a）触点输出　b）两线制 DC　c）NPN 开路集电极输出　d）NPN 电流输出
e）PNP 电流输出　f）电压电流输出

3.1.4　顺序控制执行器

1. 执行器

顺序控制执行器有电动机、电磁铁、液或气压油缸、液压电动机等。而这些执行器都可使用接触器、电磁阀进行控制。

（1）电磁铁。通电后可产生牵引力。有直流、交流驱动两种。其控制可直接用 PLC 的输出点。如果功率较大，也可经中间继电器。

（2）电动机。用以驱动运动部件。有直流电动机，更多的为交流电动机。电动机的控制主要有起动，停车及制动，调速及变向控制。这些都是用电路的切换实现。而电路切换主要用接触器。

（3）液或气压油缸、液压电动机。用以驱动运动部件。它们的控制主要有起动，停车，调速及变向控制。这些都是用油（气）路的切换实现。而油（气）路的切换主要用各种电

磁阀。

（4）电磁阀。改变通电状况，即可改变阀门位置，进而实现的油（气）通路的变化。如控制电流不大，可用 PLC 的输出触点直接控制。如控制电流超过 2A，PLC 输出点先控制中间继电器或接触器，然后通过中间继电器或接触器再控制它。

（5）电动阀用以开启、关闭阀门，以控制液（气体）体流量。它的驱动一般为电动机。所以，PLC 控制它可通过接触器。

2. 控制器

用触点直接或间接实现控制执行器的器件。触点又受它的控制线圈控制。触点有常开、常闭两种。线圈通电，触点接通，断电，触点分断，称常开触点；线圈通电，触点分断，断电，触点接通，称常闭触点。

（1）中间继电器。也称电磁式继电器，如图 3-11 所示，它由返回弹簧、铁心、线圈、衔铁、动触点、常开触点、常闭触点组成。只要在线圈两端加上额定电压、流过额定电流，所产生的电磁效应，衔铁就会在电磁力吸引下克服返回弹簧拉力吸向铁心，从而使常开触点与动触点吸合，常闭触点与动触点分断。当线圈断电，电磁吸力随之消失，衔铁就会在弹簧力作用下返回原来位置，使常开触点分断，常闭触点吸合。

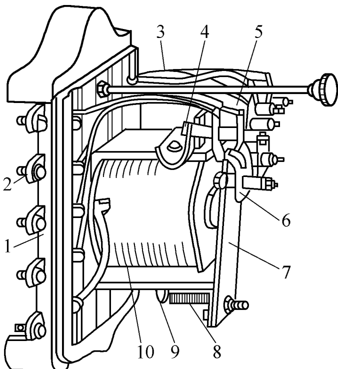


图 3-11 电磁继电器结构示意

1—底座 2—接线端子 3—连接线 4—常闭触点 5—常开触点 6—动触点 7—衔铁 8—返回弹簧 9—电磁铁 10—线圈

控制中间继电器的线圈驱动功率不大，而它所控制的触点耐压大、可通过的电流也大。因而可起到功率放大的作用。但中间继电器触点通过的电流还是小，耐电压也低，没有灭弧装置，故只能用于实现中间控制或辅助控制。中间继电器电气图形符号如图 3-12 所示。

（2）接触器。其结构如图 3-13 所示。它的工作原理与中间继电器相似。但增加有触点灭弧装置。触点可通过的电流大，耐电压高。除了主触点，还有辅助触点。其主触点可用于电动机的种种控制。辅助触点可提供反馈输入。接触器的电气图形符号如图3-14所示。

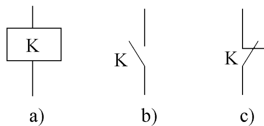


图 3-12 中间继电器图符号

a）线圈 b）常开触点 c）常闭触点

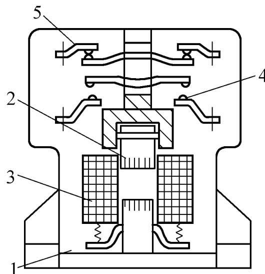


图 3-13 接触结构简图

1—铁心 2—衔铁 3—线圈 4—常开触点 5—常闭触点

（3）固态继电器（SOLID STATE RELAYS，SSR）。问世于 20 世纪 70 年代，其外观如图 3-15 所示。有两个输入端，两个输出端，中间采用隔离器件实现输入输出的电隔离。它利

用电子元件（如开关晶体管、双向晶闸管等半导体器件）开关特性，用输入端去控制输出端电路的接通和断开，因此又被称为“无触点开关”。

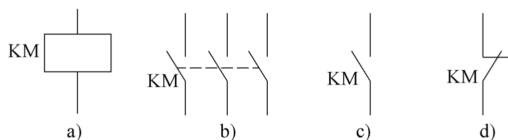


图 3-14　接触器电气图符号

a）线圈　b）主触点　c）常开触点　d）常闭触点

固态继电器类型较多。按工作性质分有直流输入-交流输出型，直流输入-直流输出型，交流输入-交流输出型，交流输入-直流输出型。按安装方式有装置式（面板安装）、线路板安装型。按元件分有普通型和增强型。

固态继电器的优点是，多数产品具有零电压导通，零电流关断，与逻辑电路兼容（TTL、DTL、HTL）切换速度快、无噪声、耐腐蚀、抗干扰、寿命长、体积小，能以微小的控制信号直接驱动大电流负载等。缺点是，存在通态压降，需要散热措施，有输出漏电流，交直流不能通用，触点组数少，成本高。

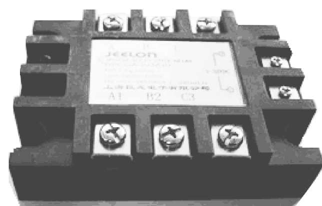


图 3-15　固态继电器外观图

由于固态继电器的内在特点，自问世以来已进入电磁继电器的大多数领域，在少数领域以完全取而代之。目前固态继电器已被广泛应用于如电炉加热系统，遥控机械、电动机、电磁阀以及信号灯、闪烁器、舞台灯光控制系统、医疗器械、复印机、洗衣机、消防保安系统等都有大量应用。

提示：这里只是对控制器的定性介绍。实际控制器的品种、型号、规格非常之多，可按需要选用。

3. 输出接线

（1）继电器输出点。没有极性，使用交流、直流电源均可。只要在触点耐压及最大通过电流限制之内，可任意连接电源与负载。直接控制负载工作。如果超出限制，可通过中间继电器或接触器再驱动负载。特别电动机这样负载更是这样。

（2）半导体输出点。有 NPN 型与 PNP 型。有极性，只能使用直流电源。而且电源的规格及型的使用也要相符。它通过的电流比继电器输出点小。但它的相应速度快。如功率不足，可利用固态继电器或半导体功率放大器放大。

（3）晶闸管输出点。只能使用交流电源。

（4）短路保护。连接负载发生短路，将烧毁输出元件及印制电路板，所以，推荐在输出电路接入保护熔丝。熔丝的容量应为输出额定值的 2 倍。

（5）使用晶体管输出时，因为存在剩余电压，故不可直接与 TTL 连接。要用 CMOS-IC 接收后，再与 TTL 单元连接。此外，晶体管输出需要用电阻来上拉。如果使用一个晶闸管输出单元去驱动低电流负载，漏电流可能使输出设备不能关断。为防止这种情况，可将一个放电电阻同负载并联。

（6）浪涌电流。使用晶体管输出，连接白炽灯等浪涌电流大的负载时，为避免损坏输出晶体管、输出晶闸管，须抑制浪涌电流。有两个方法，如图 3-16 所示。如并联，电阻可按白炽灯的暗电流为额定电流的 1/3 选定。如串联，电阻可酌情选定。

（7）感性负载。如驱动感性负载，为了防止关断时造成的电流冲击，要给感性负载并联一个浪涌抑制器或二极管，如图 3-17 所示。其元件参数选定见表 3-1。

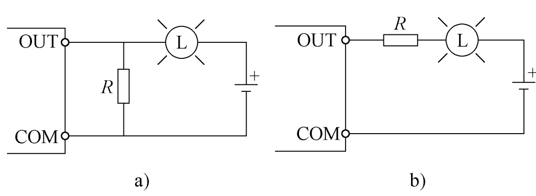


图 3-16　抑制浪涌电流方法
a）并联电阻　b）串联电阻

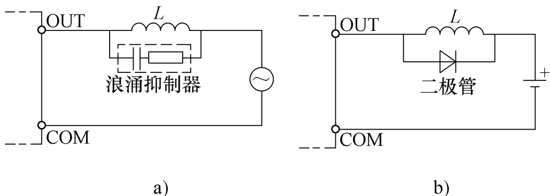


图 3-17　驱动感性负载连接处理
a）使用浪涌抑制器　b）并联二极管

表 3-1　浪涌抑制器及二极管参数

浪涌抑制器规格	二极管规格
电阻:50Ω 电容:0.47μF 电压:200V	击穿电压:至少 3 倍于负载电压,平均整流电流:1A

3.2　组合逻辑设计法编程

关键词：触点变量、触点代数、触点电路标准型、合取范式、析取范式、逻辑表达式、真值表、组合逻辑分析、组合逻辑综合

3.2.1　组合逻辑表达式与真值表

1. 表达式

继电电路常用到触点，触点代数研究的变量就是这个触点。它是研究触点电路的重要工具，也是研究梯形图逻辑的基础。

（1）有关约定

1）触点变量。继电电路常用到触点，触点变量就是反映触点状态的逻辑量。仅有两个取值，1 与 0。1 代表通，或 ON；0 代表断，或 OFF。

在一个电路中，同名器件有时用它的常开触点、有时用它的常闭触点。常开触点，没有外作用时，是断开的，有外作用时，是接通的。直接用变量名命名，如 X，读 X。常闭触点，没有外作用时，是接通的，有外作用时，是断开的。用变量名加一小横线命名，如 $\overline{X}$，读“X 的非”。可知，常开与常闭触点，其取值是相反的。

PLC 的梯形图没有实际触点，用的是操作数“位”。直接用（读）它，相当于使用常开触点。用（读）它的“非”，相当于使用常闭触点。

2）触点代数运算。触点代数是用指定运算反映触点间的连接。触点并联的运算是“或”，也叫加（+）、或析取。对应梯形图指令就是“OR”。触点串联的运算是“与”，也叫乘（*，有时乘号省略），或合取。对应梯形图指令就是“AND”。触点串联后的并联，则是乘后和。并联后的串联，则是和后乘。为了明确运算顺序，可使用成对的括号，括号内的运算优先。

此外，还有“非”的运算，也叫求反。上述同名变量的常开、常闭触点间就是“非”的关系。对常开触点求反，即变为常闭触点；对常闭触点求反即变为常开触点；求两次反，

又变为自身了。

对应梯形图，如果指令后加“NOT”，则用变量的非。如“AND　NOT　X”，是对变量 X 求反后再“与”。再如“OR　NOT　X”，是对变量 X 求反后再“或”。

触点代数未对线圈作约定，不讨论输出对输入的反馈。

（2）表达式与电路对应关系。有了上述约定，实际电路与触点代数表达式之间就有了一一对应关系。以本书第 1 章第 1 节的图 1-1、图 1-2 电路为例，除了其中的桥式电路，其它的都可用上述约定的逻辑式子表达。

L = X1 * X2（与图 1-1a 对应）

L = X1 + X2（与图 1-1b 对应）

L = （X1 * X2）+（X3 * X4）（与图 1-1c 对应）

L = （X1 + X2）*（X3 + X4）（与图 1-1f 对应）

L = （（X1 * Y1）+（X1 * Y1））*（（X2 * Y2）+（X2 * Y2））（与图 1-2b 对应）

这里，等式右边为逻辑表达式，反映了触点间的不同连接。左边为输出变量，也是用电器 L。它的取值由表达式及变量的取值决定，反映电路的效果。

图 3-2 梯形图程序为组合逻辑。也可根据上述约定用逻辑式子表达。具体为

$WW1 = XK1 \cdot XK2$

$WW2 = \overline{XK1} \cdot XK2$

$WW3 = XK1 \cdot \overline{XK2}$

（3）触点运算规则、规律及原则。数与数运算规则有：

$\overline{1} = 0$　　$\overline{0} = 1$

$0 + 0 = 0$　　$0 + 0 = 0$

$0 + 1 = 1$　　$0 + 1 = 0$

$1 + 0 = 1$　　$1 + 0 = 0$

$1 + 1 = 1$　　$1 + 1 = 1$

上述诸式是对常数的运算的规则。不难证明，这些规则是符合上述约定的。

把上述约定与命题代数对比发现，触点代数与命题代数是等价的。“这个触点接通”、“这个触点断开”就是命题，“这个触点接通或那个触点接通”就是命题运算，等等。所以，可以把命题代数运算的一些规律移植过来。这些规律有：

交换律　$X \cdot Y = Y \cdot X$

　　　　$X + Y = Y + X$

结合律　$(X \cdot Y)Z = X(Y \cdot Z)$

　　　　$(X + Y) + Z = X + (Y + Z)$

吸收律　$0 + X = X$　$1 \cdot X = X$

　　　　$1 + X = 1$　$0 \cdot X = 0$

重复律　$X \cdot X \cdot X \cdots X = X$

　　　　$X + X + X + \cdots + X = X$

分配律　$X(Y + Z) = XY + XZ$

　　　　$X + YZ = (X + Y)(X + Z)$

逆项吸收律　$X \cdot \overline{X} = 0$

$$X+\overline{X}=1$$

双重否定律　$X=\overline{\overline{X}}$

反演律　$\overline{X+Y+Z}=\overline{X}\cdot\overline{Y}\cdot\overline{Z}$

$$\overline{XYZ}=\overline{X}+\overline{Y}+\overline{Z}$$

这些规律也完全可用实际电路予以验证。

此外，还有 3 条原则：

1）反演原则。它是反演律的推广。任何逻辑式子 F，若“非”仅出现在变量上，把这个公式中的乘与加对换、变量与变量的非对换、1 与 0 对换，所得的式子 G，满足

$$G=\overline{F}$$

利用它，很易求出与某一电路功能相反的电路。

2）对偶原则。任一逻辑式子 F，若非仅出现在变量上，把这个公式的乘与加对换、0 与 1 对换，所得的式子 F′称其为 F 的对偶公式。对偶是相互的，F 也是 F′的对偶公式。

若有 $F=G$

则它的任一对偶 F′、G′有

$$F'=G'$$

有了这个原则，等式的证明可减少一半。

3）代入原则。任何含有变量 A 的逻辑式子，如果把所有出现 A 的地方都代之以一个逻辑式子 F，则等式仍然成立。这是因为一个逻辑式子的取值，也和逻辑变量一样，不是 1 就是 0，所以，这个原则是正确的。但一定要注意，这个“所有”和“都”，不能部分地被代替，那样就错了。

这些规律、原则是很有用的。可用以化简逻辑表达式。对应电路、梯形图，就可用最少触点、指令，去表达相同的逻辑关系。

提示：本书第 1 章中图 1-1 及图 1-2 中若干功能相同的电路都可运用上述规律、原则予以证明。

（4）触点电路范式。从工艺与使用上考虑，触点电路应标准化。使用 PLC 的指令也有这个问题。标准化的触点电路称触点连接标准型，也称范式。有两种范式：合取范式与析取范式。

1）合取范式。它是由一些析取范式的合取，是先加后乘的逻辑式。其对应的电路是先并、后串。如表达式

$$(X+Y)*(Y+Z)*(Z+X)$$

即为合取范式。

容易证明，任一触点电路的逻辑式，运用上面讲的规律、原则作一变换，总可以化为合取范式。如表达式：

$$\begin{aligned}&X*Y+Y*Z+Z*X\\&=(X+Y)*(X+Z)*(Y+Y)*(Y+Z)+(Z*X)\\&=(X+Y)*(X+Z)*(Y)*(Y+Z)+(Z*X)\\&=(X+Y+Z)*(X+Y+X)*\\&\quad(X+Z+Z)*(X+Z+X)*(Y+Z)*\\&\quad(Y+X)*(Y+Z+Z)*(Y+Z+X)\\&=(X+Y)*(Y+Z)*(Z+X)\end{aligned}$$

这里用了分配律作变换，把不是析取范式变换成合取范式了。也可运用反演律求解。其结果也是一样的。

运用合取范式很容易求出代数式为零的条件。如上式，只要在 3 个和项中，有一个为零，这个式子的结果即为零。

2）析取范式。它由一些合取范式的析取组成，是先乘后加的式子。其对应的电路是先串、后并。梯形图很常用这个范式。如表达式

$$X*Y+Y*Z+Z*X$$

即为析取范式。

也容易证明，任何一个触点电路的逻辑式，运用上面讲的规律、原则作变换，总可以化为析取范式。如

$$\begin{aligned}&(X+Y)*(Y+Z)*(Z+X)\\&=(X*Y+X*Z+Y*Y+Y*Z)*(Z+X)\\&=(X*Y+X*Z+Y+Y*Z)*(Z+X)\\&=X*Y*Z+X*Y*X+X*Z*Z+X*Z*X\\&+Y*Z+Y*X+Y*Z*Z+Y*Z*X\\&=X*Y+Y*Z+Z*X\end{aligned}$$

这里用了分配律、吸收律、结合律等作了变换，把合取范式变成了析取范式。也可运用反演律求解。其结果也是一样的。

用析取范式很容易求出逻辑式为 1 的条件，如上式，只要在 3 个乘积项中，有一个为 1，这个式子即为 1。

应该指出，范式虽是标准型的触点连接，但它不是惟一的。逻辑化简，就是要寻找最简的范式，即寻找能实现所要求的功能而又是标准化型触点连接的、且用的触点又是最少的逻辑式子。这对 PLC 讲，就是用最少的指令，去实现所要求的功能。

（5）触点电路特异范式。由于范式不是惟一的，不大好把握它，为此再介绍一下特异范式。

1）1 的组分。1 作为逻辑量，可用下式表达，即

$$1=(X_1+\overline{X}_1)(X_2+\overline{X}_2)\cdots(X_n+\overline{X}_n)$$

根据分配律，可把上式的右边展开成析取范式，共 2 的 n 次方项，即

$$X_1X_2\cdots X_n+X_1X_2\cdots\overline{X}_n+\cdots+\overline{X}_1\overline{X}_2\cdots\overline{X}_n$$

若把以上乘积项中，X_i 看成 1，$\overline{X}_1$ 看成 0，则其每一项均可看成是一个二进制的数。然后再对应地译成十进制数，则有

$$\begin{aligned}R_0&=\overline{X}_1\overline{X}_2\cdots\overline{X}_n\\R_1&=\overline{X}_1\overline{X}_2\cdots X_n\\R_n&=\overline{X}_1\overline{X}_2\cdots X_{n-1}\overline{X}_n\\&\cdots\\R_{2^n-1}&=X_1X_2\cdots X_n\end{aligned}$$

这样，表达 1 的式子可写成

$$1 = R_0 + R_1 + \cdots + R_{2^n-1} = \sum^{2^n-1} R_i$$

这里的 R_i 就叫做 1 的组分，1 是由它的所有组分的析取而成。

显然，1 的组分要随所讨论的变量多少而变化。当变量数确定后，它的组分是确定的。

可以证明，任何一个触点电路，除恒不通外，表达它的触点逻辑式总可以展开成若干 1 的组分的合取。如

$$\begin{aligned} XY + X\overline{Z} &= XY(Z+\overline{Z}) + X(Y+\overline{Y})\overline{Z} \\ &= XYZ + XY\overline{Z} + XY\overline{Z} + X\overline{Y}\,\overline{Z} \\ &= XYZ + XY\overline{Z} + X\overline{Y}\,\overline{Z} = R_7 + R_6 + R_4 \end{aligned}$$

1 的组分又称最小项目，上式有 3 个 1 的组分，即说它包含了 3 个最小项。

最小项表达的逻辑式子又叫特异析取式。显然，对一个具体的电路，变量数确定之后，特异析取式是惟一的。

与 1 分解为 n 个变量的组分的析取相对应的电路叫触点塔。图 3-18 所示为 3 个变量的触点塔。

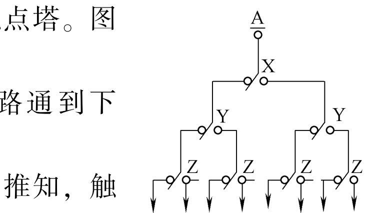


图 3-18　三个变量的触点塔

从图知，不管 X、Y、Z 怎么搭接，A 点总有一条路通到下面来。

这个触点塔有 3 层，因为它含有 3 个变量。由此可以推知，触点塔的层数总是等于它所含的变量数。

因为任何一个触点电路，总可以用 1 的若干组分的析取表达，所以，它也总可以看成是 1 的触点塔的某一部分。反之，有目的地取出 1 的触点塔的一部分，总可以构成满足一定要求的电路。

2）0 的组分。与 1 的组分相对应的，还有 0 的组分。根据反演律，有

$$0 = \overline{1} = \overline{\sum_0^{2^n-1} R_i} = \overline{R_1} \cdot \overline{R_2} \cdots \overline{R_{2^n-1}}$$

其中 R_i 可展开成和的式子，并称之为 0 的组分。如 $n=2$，则

$$\begin{aligned} 0 &= \overline{R_0} \cdot \overline{R_1} \cdot \overline{R_2} \cdot \overline{R_3} \\ &= (X_1 + X_2)(X_1 + \overline{X_2})(\overline{X_1} + X_2)(\overline{X_1} + \overline{X_2}) \end{aligned}$$

其对应的电路如图 3-19 所示。它为多节构成的，n 个变量有 2^n 个节，每节有 n 个节点并联。从图可看出，不管怎么搭接，A 到 B 总是断的。这正好与它用来表达 0 相对应。

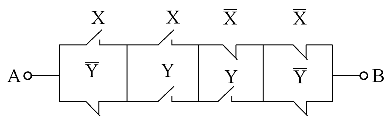


图 3-19　多节构成电路

也可证明，任何触点电路，除为恒通外，均可以变换成若干 0 组分的合取。这个合取式也称该电路的特异合取式。显然，对任一具体电路，变量数一定时，表达它的特异合取式是惟一的。

既然任一具体电路可表达成特异合取式，那么，取出 0 电路的一部分（即去掉若干节），只要它符合特异合取式，则它也完全可实现具体电路的功能。

两种特异式子可按一定规律进行转换。具体是，特异合取式是 0 的组分先减去要转换的特异析取式的所有项，再把余下的各项分别取非，然后再合取；特异析取式是 1 的组分先减去要转换特异合取式的所有的项，再把余下的各项分别取非，然后再析取。如特异析取式

$$XY\overline{Z}+X\overline{Y}Z+\overline{X}\,\overline{Y}Z+X\overline{Y}\,\overline{Z}$$

求其等效特异合取式，可：先求 1 的组分，再减去上式中所含的 1 组分，余下的为

$$XYZ、\overline{X}\,\overline{Y}\,\overline{Z}、\overline{X}YZ、\overline{X}Y\overline{Z}$$

对这四项再分别取非，各为

$$\overline{XYZ}=\overline{X}+\overline{Y}+\overline{Z}$$

$$\overline{\overline{X}\,\overline{Y}\,\overline{Z}}=X+Y+Z$$

$$\overline{\overline{X}YZ}=X+\overline{Y}+\overline{Z}$$

$$\overline{\overline{X}Y\overline{Z}}=X+\overline{Y}+Z$$

以上四项的积，即为这里所求的特异合取式，即

$$(\overline{X}+\overline{Y}+\overline{Z})(X+Y+Z)(X+\overline{Y}+\overline{Z})(X+\overline{Y}+Z)$$

反之，也可倒推回去，其推导过程略。

从以上介绍可知，变量数定了之后，对任一电路，表达它的特异范式总是确定的、惟一的。而且两种特异范式可以互相转换。特异范式用的触点多，实际上不用它，但可说明使用触点最多时的情况，对把握与理解电路是很有用的。

（6）多输出触点电路及其数学表达式。以上讨论触点电路都只有一个输出值，是单输出触点电路。实际电路有时要求同时有多个输出变量，如要对多个对象控制，就要用到多输出触点的电路。若用数学表达式表示多输出触点的电路，则为一组表达式，表达式的右边不仅无自身变量，而且也无其它输出变量。具体如下：

$$y_1=f_1(x_1,x_2,\cdots,x_n)$$

$$\vdots$$

$$y_m=f_m\ (x_1,\ x_2,\ \cdots,\ x_n)$$

这里有 n 个输入，即 x_1，x_2，…，x_n，同时还含有它们的非。m 个输出，即 y_1，y_2，…，y_m，同时还含有它们的非。对应的有 m 个表达式。图 3-20 表示的组合逻辑的入出关系。

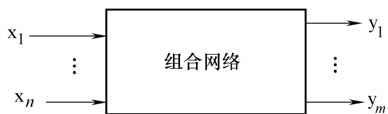


图 3-20　组合逻辑的入出关系

上述表达式也称为组合逻辑的一般表达式。

多输出电路的化简可先分别对各个输出的表达式进行化简，然后再作输入触点兼用“合并”处理，以求在总体上用最少的输入触点，实现多触点输出。有关多输出触点电路化简详细介绍可参阅有关多输出组合逻辑的专著。

2. 真值表

真值表是由行与列组成，用以记录输入变量与输出间的对应关系。它的“列”记录着变量的不同取值；“行”记录着输入变量不同取值时，输出的取值。表中 1 代表器件工作，常开触点 ON，常闭触点 OFF；0 代表器件不工作，常闭触点 ON，常开触点 OFF。

如表 3-2，就是反映 A、B 两个输入变量不同取值时，它对应的输出，做不同逻辑运算后的值。

提示：真值表的输入一般应涵盖所有的可能，但顺序可任意。

3.2.2　组合逻辑分析

是对已存在触点电路可能输出进行求解，以弄清该逻辑电路可能实现的功能。

表 3-2　A、B 两触点不同逻辑运算的真值表

输入		输　　出							
A	B	$\overline{A}$	$\overline{B}$	A · B	A + B	$\overline{A \times B}$	$\overline{A + B}$	$A+\overline{B}$	$A \cdot \overline{B}$
0	0	1	1	0	0	1	1	1	0
0	1	1	0	0	1	1	0	0	0
1	0	0	1	0	1	1	0	1	1
1	1	0	0	1	1	0	0	1	0

分析的办法是：根据实际电路，或梯形图，或指令关系，按触点代数的约定，列出逻辑表达式；用以上介绍的方法，对表达式进行化简；把输入变量的各种可能取值代入化简后表达式，求出相应的输出值；根据求得的输入、输出对应关系，弄清该逻辑电路可能实现的功能。

以下以图 3-2 所示的刀架位置显示程序为例，介绍组合逻辑的分析，从该图中可知：

$$WW1 = XK1 \cdot XK2$$
$$WW2 = \overline{XK1} \cdot XK2$$
$$WW3 = XK1 \cdot \overline{XK2}$$

而 XK1、XK2 可能的取值只能是 11、01 及 10。运用上述表达式，对应的 WW1、WW2 及 WW3 的值进行运算，其结果分别为 100、010 及 001，见表 3-3。

表 3-3　图 3-2 的真值表

输　　入		输　　出		
XK1	XK2	WW1	WW2	WW3
1	1	1	0	0
0	1	0	1	0
1	0	0	0	1

从表中可知，用 WW1、WW2 及 WW3 三个指示灯显示刀架的不同位置，是完全可行的。

3.2.3　组合逻辑综合

综合是分析的反问题，是根据功能去设计电路。其答案往往不是一个。还要优化答案，比其分析要复杂些。

为此，综合要有算法。其步骤是：

分析设计要求，设计真值表，把要求抽象为输入与输出间的对应的逻辑关系；

依据真值表列写逻辑表达式；

表达式化简；

按约定，画出触点电路，即梯形图程序。

逻辑化简，是在保持逻辑关系不变的前提下，简化表达式。其标准是，对触点电路，总触点数要少；对 PLC 程序，使用的指令数量最少。

化简的方法很多，除了用上述规律、原则化简，还有几何法（卡诺图）、Q—M 法等等。

也可用计算机辅助化简。若按最简析取范式的目标化简，其本质都是先求出最大的蕴涵项，然后再从中挑选一组既最简，又包含所有最小项的最大蕴涵项，然后再对其析取。所得最大蕴涵项是指这样的乘积项，它所可能包含的构成被化简的逻辑式子的最小项最多。

有了化简后的逻辑表达式，进而画出对应的电路图，就是指令及地址的选用，是比较好处理的。

3.2.4　组合逻辑综合实例

1. 三个钮子开关控制一个灯的逻辑

要求是，有三处安装有钮子开关。任何一处均可改变灯的状态。如灯亮，可使其灭；反之，可使其亮。

（1）设计真值表。设 3 个开关分别用 A、B 及 C 表示。灯用 L 表示。每个开关都有两个状态，即下扳、上扳。下扳时用变量本身，即用 A、B 或 C 表示。上扳时用变量的非，即用 $\overline{A}$、$\overline{B}$及$\overline{C}$表示。

这 3 个变量各有两个取值，其可能的组合有 8 种。把这 8 种分成两组，奇数个下扳的一组，有 4 个；偶数个下扳的另一组，也有 4 个。显然，任何一个开关状态的改变，都将组合从一组改变到另一组。如果用其中一个组合使灯亮，另一个组合使灯灭，即可实现所要求的控制了。

对此分析，可用真值表表示。见表 3-4。

表 3-4　一灯多控制真值表

A	B	C	L	A	B	C	L
1	1	1	1	1	0	1	0
1	0	0	1	1	1	0	0
0	1	0	1	0	1	1	0
0	0	1	1	0	0	0	0

提示：这里 A=0 意味着$\overline{A}$=1，B=0 意味着$\overline{B}$=1，C=0 意味着$\overline{C}$=1。故在表中，变量非均不必列出。

（2）列写表达式。

$$L = ABC + A\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}\,\overline{B}C$$

当然还应对这个表达式进行化简。不过由于使用 PLC，触点多少问题不大，可直接依据此表达式，画出的梯形图，如图 3-21 所示。图 3-21a 为其合取范式，即先串、后并。图

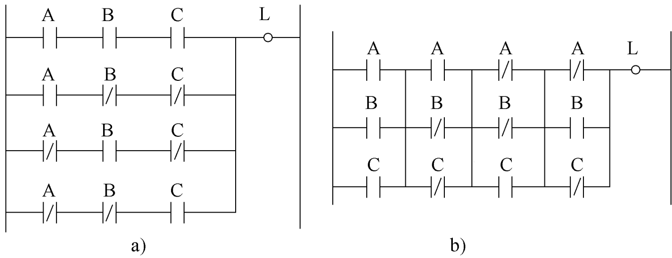


图 3-21　三个钮子开关控制一个灯电路

a）合取范式　b）析取范式

3-21b 为其析取范式，即先并、后串并。

这里仅用 3 个开关管理一个灯。如果多了，如几十、几百，就太复杂了。为此可用后面将要介绍的高级逻辑设计方法设计。

2. 多开关表决逻辑

本例用 3 个开关，2 个赞成，表示通过。

（1）设计真值表。3 个开关分别用 A、B 及 C 表示。表决结果用灯 L 表示。每个开关都有两个状态，即下扳（赞成）、上扳（不赞成）。下扳时用变量本身，即 A、B 或 C 表示。上扳时用变量的非，即$\overline{A}$、$\overline{B}$及$\overline{C}$表示。这 3 个变量各有两个取值，下扳的是两个及两个以上的有 4 种。把这 4 种组合使灯亮，其它组合使灯灭，即可实现所要求的控制了。

对此分析，可用真值表表示，见表 3-5。

表 3-5　表决控制真值表

A	B	C	L	A	B	C	L
1	1	1	1	1	0	0	0
1	1	0	1	0	1	0	0
0	1	1	1	0	0	1	0
1	0	1	1	0	0	0	0

提示：这里 A = 0 意味着$\overline{A}$ = 1，B = 0 意味着$\overline{B}$ = 1，C = 0 意味着$\overline{C}$ = 1。故在表中，变量非均不必列出。

（2）列写表达式。

$$L = ABC + AB\overline{C} + \overline{A}BC + A\overline{B}C$$

经化简则为

$$\begin{aligned} L &= ABC + AB\overline{C} + \overline{A}BC + A\overline{B}C \\ &= ABC + AB\overline{C} + ABC + \overline{A}BC + ABC + A\overline{B}C \\ &= AB + BC + AC \\ &= (A + B)(B + C)(C + A) \end{aligned}$$

当然还应对这个表达式进行化简。不过由于使用 PLC，触点多少问题不大，可直接依据此表达式，画出的梯形图，如图 3-22 所示。图 3-22a 为其合取范式。图 3-22b 为其析取范式。这个电路在以上讨论等效输出时也见过。只是那里用了直接地址。

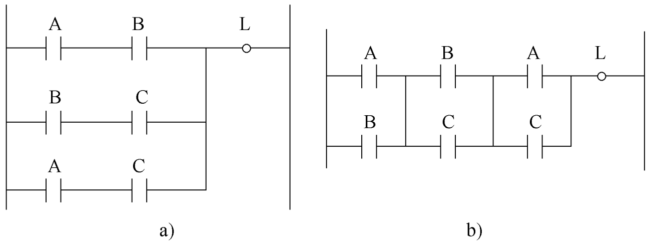


图 3-22　三个开关表决电路

a）合取范式　b）析取范式

这里仅用 3 个开关表决。如果多了，如几十、几百，就太复杂了。为此可用后面将要介

绍的高级逻辑设计方法设计。

3. 比较逻辑

比较甲、乙两组开关上扳、或下扳是否一致。这里假设每组有 3 个开关。

（1）设计真值表。甲组的 3 个开关分别用 A、B 及 C 表示，乙组与甲组对应的 3 个开关分别用 X、Y 及 Z 表示。比较相同结果用灯 L 表示。每个开关也都有两个状态，即下扳、上扳。下扳时用变量本身表示。上扳时用变量的非表示。

这 3 个变量各有两个取值。以 A 与 X 为例，如比较一致，要不都是上扳，要不都是下扳。即：$AX+\overline{A}\,\overline{X}$。

而只有 3 对比较都一致，才说明这两组开关一致。以上分析见表 3-6。

表 3-6　比较控制真值表

A	X	B	Y	C	Z	L
1	1	1	1	1	1	1
1	1	0	0	0	0	1
1	1	0	0	1	1	1
1	1	1	1	0	0	1
0	0	1	1	1	1	1
0	0	0	0	1	1	1
1	1	0	0	1	1	1
1	1	0	0	1	1	1
0	0	0	0	0	0	1

提示：这里 $A=0$ **意味着**$\overline{A}=1$**，**$B=0$ **意味着**$\overline{B}=1$**，**$C=0$ **意味着**$\overline{C}=1$。X、Y、Z **也类似。故在表中，变量非均不必列出**。

（2）列写表达式。

$$L=(AX+\overline{A}\,\overline{X})(BY+\ +\overline{B}\,\overline{Y})(CZ+\overline{C}\,\overline{Z})$$

当然还应对这个表达式进行化简。不过由于使用 PLC，触点多少问题不大，可直接依据此表达式，画出的梯形图，如图 3-23 所示。图 3-23a 为其析取范式。图 3-23b 为其合取范式。这里变量声明略。

这里仅用两组 3 个开关比较。如果多了，如几十、几百，就太复杂了。为此可用字节、字或双字比较指令了。

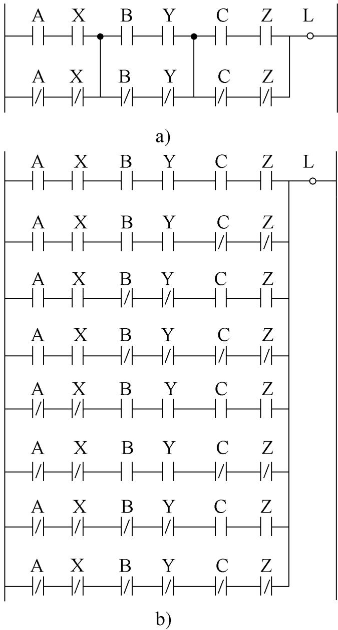


图 3-23　3 开关比较逻辑

a）析取范式　b）合取范式

4. 格兰码到二进制码译码

格兰码为单位码，不少绝对值计数的旋转编码器用它编码。但格兰码没有“权值”，无法用作大小比较。但它与二进制码有对应关系，其关系真值表见表 3-7（5 位）。以此关系，可把它译成二进制码。

从表 3-7 可知，二进制码的本位的值为：格兰码的本

表 3-7　二进制码与格兰码对照真值表

格　兰　码					序号	二　进　制　码				
0	0	0	0	0	00	0	0	0	0	0
0	0	0	0	1	01	0	0	0	0	1
0	0	0	1	1	02	0	0	0	1	0
0	0	0	1	0	03	0	0	0	1	1
0	0	1	1	0	04	0	0	1	0	0
0	0	1	1	1	05	0	0	1	0	1
0	0	1	0	1	06	0	0	1	1	0
0	0	1	0	0	07	0	0	1	1	1
0	1	1	0	0	08	0	1	0	0	0
0	1	1	0	1	09	0	1	0	0	1
0	1	1	1	1	10	0	1	0	1	0
0	1	1	1	0	11	0	1	0	1	1
0	1	0	1	0	12	0	1	1	0	0
0	1	0	1	1	13	0	1	1	0	1
0	1	0	0	1	14	0	1	1	1	0
0	1	0	0	0	15	0	1	1	1	1
1	1	0	0	0	16	1	0	0	0	0
1	1	0	0	1	17	1	0	0	0	1
1	1	0	1	1	18	1	0	0	1	0
1	1	0	1	0	19	1	0	0	1	1
1	1	1	1	0	20	1	0	1	0	0
1	1	1	1	1	21	1	0	1	0	1
1	1	1	0	1	22	1	0	1	1	0
1	1	1	0	0	23	1	0	1	1	1
1	0	1	0	0	24	1	1	0	0	0
1	0	1	0	1	25	1	1	0	0	1
1	0	1	1	1	26	1	1	0	1	0
1	0	1	1	0	27	1	1	0	1	1
1	0	0	1	0	28	1	1	1	0	0
1	0	0	1	1	29	1	1	1	0	1
1	0	0	0	1	30	1	1	1	1	0
1	0	0	0	0	31	1	1	1	1	1

位值与二进制码高一位值的异或，即：

$$g(i)=(\overline{g(i)})(e(i+1))+(g(i))(\overline{e(i+1)})$$

这里　$e(i)$ 为第 i 位二进制值；

　　　$g(i)$ 为第 i 位格兰码值；

$e(i+1)$ 为第 $i+1$ 位二进制值。

而最高位两者相等。

图 3-24 即为此译码程序。图 3-24a 为其合取范式。图 3-24b 为其析取范式。该图 g0（低位）到 g4（高位）为格兰码，e0（低位）到 e4（高位）二进制码。

5. 二进制码到格兰码译码

从表 7-5 中可知，格兰码本位的值为二进制码的高一位值与二进制码的本位值的异或，即：

$$g(i)=(\overline{e(i)})(e(i+1))+(e(i))(\overline{e(i+1)})$$

这里　$e(i)$ 为第 i 位二进制值；

$e(i+1)$ 为第 $i+1$ 位二进制值；

$g(i)$ 为第 i 位格兰码值。

而最高位两者相等。

如图 3-25 所示的为二进制到格兰码的译码程序，图 3-25a 为其合取范式，图 3-25b 为其析取范式。该图 g0（低位）到 g4（高位）为格兰码，e0（低位）到 e4（高位）为二进制码。

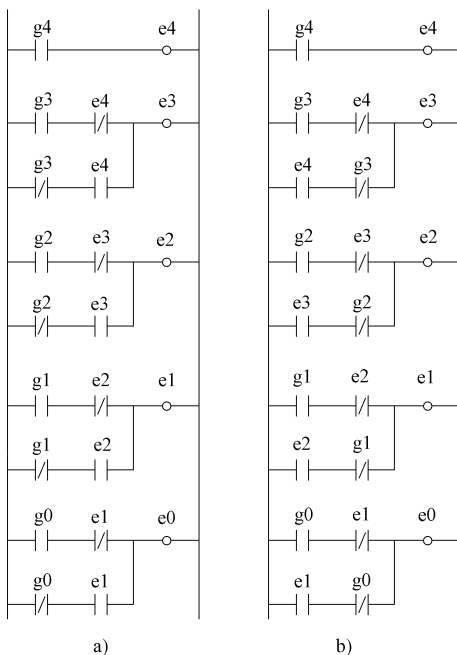


图 3-24　格兰码到二进制译码程序

a）合取范式　b）析取范式

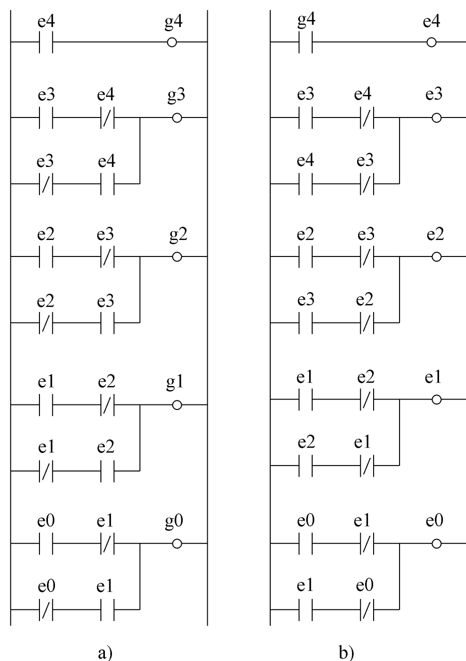


图 3-25　二进制到格兰码译码程序

a）合取范式　b）析取范式

3.3　异步时序逻辑编程

关键词：分析、综合、梯形图逻辑、步、节拍、主令信号、反馈信号、有效输入、长信号、短信号、计数信号、通电表

PLC 顺序控制，其变量间的关系几乎都是时序逻辑关系。PLC 指令是一条一条执行的，

先执行的指令执行后，所引起的输出变量或内部状态变量变化，将可能改变后执行指令的执行条件。同时，PLC 指令还是不间断地周期执行，前一扫描周期执行的结果，也将对次一周期的执行产生影响。所以，它是典型的异步时序逻辑。

本节将介绍异步时序逻辑编程及与其有关的梯形图逻辑。

3.3.1 异步时序逻辑表达式与通电表

1. 表达式

时序逻辑要用到线圈及其控制触点的反馈。故还应对此作约定。具体为，线圈与受其控制的触点名称相同，线圈工作或 ON，用 1 表示。这时，其常开触点接通，常闭触点断开。线圈不工作或 OFF，也用 0 表示。这时，其常闭触点接通，常开触点断开。

PLC 没有线圈及受其控制的触点，用的是操作数“位”的写与读。如果调 OUT 指令，并用 1 写操作数“位”，相当于使线圈工作。这时，如直接用（读）它，相当于使用常开触点，则 ON；如用（读）它的“非”，相当于使用常闭触点，则 OFF。如果用 0 写操作数“位”，相当于使线圈不工作。这时，如直接用（读）它，则 OFF；如用（读）它的“非”，则 ON。

操作数“位”也可调 OUT—NOT 指令写它。它与 OUT 为“非”的关系。OUT—NOT 指令用 0 写，相当于 OUT 指令用 1 写。OUT—NOT 指令用 1 写，相当于 OUT 指令用 0 写。

梯形图逻辑的运算规则、规律及化简原则与触点代数相同。

（1）梯形图逻辑基本表达式。有了上述约定，可将实际梯形图用逻辑式子表达。如本书第 1 章讨论得起、保、停梯形图（见图 1-13），其表达式为

$$10.00=(0.00+10.00)*\overline{0.01}$$

这里，等式左边的是线圈 10.00。而等式右边的（10.00）为线圈 10.00 控制的常开触点，是对线圈工作的反馈控制。

对图 3-3 的各个“条”也可列写逻辑表达式。如“条”0、1 为

$$WW1=XK1*MM3+\overline{MM1}*WW1$$

$$MM1=\overline{XK1}*WW1+\overline{WW2}*MM1$$

其它“条”也类似。

（2）梯形图逻辑一般表达式。一般讲，任一线圈 Ji，仅用 LD、OUT、AND、OR、AND—NOT、OR—NOT、AND—LD、OR—LD 等基本指令的操作数控制时，其逻辑表达式为

$$J_i=f_i(a_1、a_2、\cdots、a_j\cdots、a_n,J_1、J_2、\cdots、J_k、\cdots、J_m)$$

式中 a_j——输入继电器的触点，可能为常开，也可能常闭；

J_k——内部辅助、输出继电器触点，可能为常开，也可能常闭。

可以不用证明地指出，上式可以分解成两组，一组为含有自身触点这个因子，另一组不含有自身触点这个因子。即：

$$Ji=Qi+BiJi \tag{3-1}$$

式中，Qi、Bi 为都不含 Ji 的因子的逻辑式子。从硬件意义上讲，式左边的 Ji 代表线圈，右边 Ji 代表它的触点。

其对应的梯形图如图 3-26 所示。

读者可能要问，怎么没有 Ji 非的因子呢？主要是考虑以自身的常闭触点去控制自身的线圈，没有实际意义。不然，如图 3-27 所示，其含义是：每执行一次这组指令，Ji 的状态变换一次。无特殊需要，一般不作这样设计。

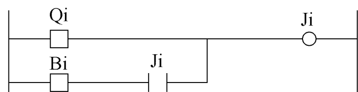


图 3-26　与式（3-1）对应的梯形图

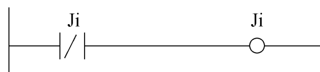


图 3-27　不稳定电路

有了这个思路，处理时序逻辑的问题，则也可用组合逻辑的方法对时序逻辑进行分析了。分析时，分起动与保持两个部分：

1）起动。看由 0 变为 1 的条件，只要变为 1，且保持也为 1，这个输出如未起动，则起动，并能保持；

2）保持。看由 1 变为 0 的条件，只要变为 0，且起动不为 1，这个输出如已起动，则停止，不再工作。

而 Qi、Bi 都是触点代数表达的逻辑式，用触点代数知识处理就可以了。

梯形图中还用有锁存指令，它分有置位（S）及复位（R）。置位实质上即为起动，与起动电路（Qi）对应，复位实质也只是保持的非，与保持电路中 Bi 的非对应。

梯形图输出还有 OUT—NOT 指令。它是 OUT 取反。也可列写表达式，但由于 OUT—NOT 指令不常用，故这里略。

（3）多输出梯形图逻辑表达式。若更一般用讨论梯形图逻辑，则可用隐函数表示时序逻辑的入处关系。具体是：

y1 = f1(x1,x2⋯xp,m1,m2⋯mq,y1,y2⋯yr)

y2 = f2(x1,x2⋯xp,m1,m2⋯mq,y1,y2⋯yr)

⋮

yr = fr(x1,x2⋯xp,m1,m2⋯mq,y1,y2⋯yr)

m1 = f1(x1,x2⋯xp,m1,m2⋯mq,y1,y2⋯yr)

m2 = f2(x1,x2⋯xp,m1,m2⋯mq,y1,y2⋯yr)

⋮

mq = fq(x1,x2⋯xp,m1,m2⋯mq,y1,y2⋯yr)

这里有 p 个输入，即 x1，x2⋯xp，同时还含有它们的非。r 个输出，即 y1，y2，⋯，yr，同时还含有它们的非。而且，输入端的 y 与输出端的 y 的取值是有时差的，入端的 y 为前一次的值。对应的有 r 个 y 表达式。

此外，还有 q 个变量表示内部状态的变量，即 m1，m2，⋯，mq,同时还含有它们的非。而且，输入端的 m 与输出端的 m 的取值是有时差的，入端的 m 为前一次的值。对应的有 q 个 m 表达式。图 3-28 表示的为时序逻辑的入出关系。

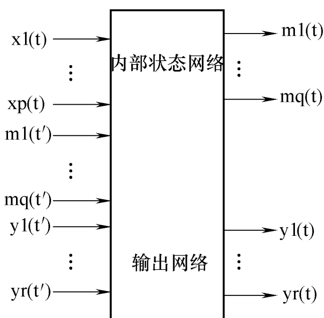


图 3-28　时序逻辑入出关系

从图知，它有两个网络：输出网络与内部状态网络。

而且，两个网络都可能有输出与内部状态变量的反馈，即它的输出又可能成为它自身的输入。只是 t 时刻的输出与内部状态的取值取决于 t′时刻的输出、内部状态的取值及 t 时刻的输入的取值。这里 t′可能（同步时序处理时）全为 t－1，也可能（异步时序）有的为 t－1，有的为 t，以至于很多中间的过渡时间，关系较复杂。

2. 通电表

异步逻辑的重要特点是讲顺序关系。为此，这里引用作者在《机床控制电器及电控制器》一书中，用以研究继电器电路的通电表。继电器电路是典型的异步时序逻辑。既然它的分析、设计能使用通电表，同样是时序逻辑的 PLC 顺序控制，当然也能用。

通电表由行与列组成，见表 3-8。它的“列”记录着各个器件（对 PLC 应是内存中的工作位、布尔变量）在各个节拍的工作状况；“行”记录着各个节拍各个器件的工作情况。表中 1 代表器件在工作，其常开触点 ON，常闭触点 OFF；0 代表器件不在工作，其常闭触点 ON，常开触点 OFF。

表 3-8　通电表

节拍序号	当前输入	输入器件 1	输入器件 2	…	内部器件 1	内部器件 2	…	输出器件 1	输出器件 2
0									
1									
2									

节拍与输入有关，它的转换也是由输入引起的。而对时序电路的输入总是作以下两个假设：

（1）在同一时间总是只有一个输入信号改变。

（2）两次输入信号改变之间系统的内部及输出状态已趋于稳定。

事实上，绝大多数工作系统是总是能满足这两个假设的。

因此，用这个表可从时、空两个角度看出时序逻辑的全貌。用“通电表”分析继电器电路，过程较清晰，不易“糊涂”，这正如用笔算比用心算不易“糊涂”的道理一样。而把它用于 PLC 顺序控制程序的分析与设计，也一样清晰，便于理解。

具体的输入信号分有主令信号与反馈信号。主令信号是由主令器（如起、停按钮）发出（对 PLC 为通过输入点）的，它主动作用于电路；反馈信号是由 PLC 输出的反馈，由检测传感器（如行程开关）发出（对 PLC 也为通过输入点）的，是对 PLC 控制动作执行后的回应，是被动的。

一般来讲，确定电路、自动控制电路主令信号较少，一个或最多两个（起、停）。但随机电路、手动控制电路可能较多，如电梯电路，各选择按钮都是主令器。

分清主令信号与反馈信号对分析与设计梯形图逻辑很有好处。因为多数电路开始工作时总是由主令信号引起的，而以后的工作推进则多是反馈信号。找出主令信号就等于抓住了开头，也就有了头绪，这样，再进一步展开分析与设计自然也就不难了。

不是所有的输入改变都会改变输出（对 PLC 为输出通道、即点），或改变内部状态（对 PLC 为中间存储区的位）。不会产生这种“改变”输入信号可视为无效信号。反之，为有效信号。如按钮，一般来讲，从松开到按下为有效信号，而从按下到松开则多为无效信号。但

这也不是绝对的。要看怎么去处理。如用一个按钮实现起、停控制，则按下与松开可能都将是有效信号。

有了有效输入信号的概念，还可对节拍的概念作更进一步说明。节拍也可认为是两个有效输入之间的时间区段。

通电表的当前输入是进入本节拍的信号。如果信号从 OFF 到 ON，用信号的名（变量本身）。如果信号从 ON 到 OFF，用信号的名上加小横线（变量的非）。

提示：通电表的当前输入不一定都是某输入信号的改变。内部器件或输出器件的状态改变也可。

通电表是按有效信号输入的顺序，对所有节拍的各个器件的状态逐一作了记载，因而可全面反映时序逻辑的工作全貌。

对有效输入信号，还可分有长信号、短信号与计数信号：

（1）如同一输入变量的正与反取值都作为相邻两次输入则为短信号。

（2）如同一输入变量的正与反取值都不作为相邻两次输入，即在它们之间夹有其它输入变量，则为长信号。

（3）如连续地用同一输入变量的正与反取值都作为多次相邻输入则为计数信号。

实际处理时，这些信号是可变动或可调整的。如用按钮起动一个动作，若按钮按下直到动作发生，并收到动作发生的反馈信号后按钮才松开，则这个按钮按信号即为长信号。若按钮按下未到动作发生，并未接收到动作发生的反馈信号按钮就松开，则这个按钮按信号即为短信号。

把信号处理成不同的特性，对其作逻辑分析或综合，结果有可能是不同的。所以，区分与了解信号的这些特性，将有助于梯形图逻辑的分析与综合。

提示：通电表与真值表不同。通电表输入可不涵盖所有的可能。但它的当前输入顺序不能是任意的，要根据实际工作过程确定。

3.3.2　异步时序逻辑分析

分析是研究设计好的程序。也可说是在“纸面”上“运行”程序。是检查程序正确性、可行性的一个方法。

具体方法是按有效输入的实际顺序，逐一分析所有器件的状态，确定是 0、OFF、不工作，还是 1、ON、工作，进而弄清程序能否实现预定的功能。

显然，如果程序正确、可行，其输入、输出的对应关系也总是确定的。也就是说，对某个具体程序的分析，其答案只有一个。

时序逻辑程序分析比较复杂、繁琐。但若从原始状态开始，抓住当前输入的逐一变化及产生的后果，按节拍逐步推进，再运用通电表进行记录，还是不难的。

1. 分析图 3-3 程序

该逻辑的主令器件为 XK1，所有节拍的转换都由它的变化引起。原始状态时为节拍 0。这时，XK1 压下，ON，WW1 ON。显示处位置 0。

节拍 1：在节拍 0，刀架转动，XK1 松开、OFF，进入此节拍。这时，MM1 ON，而 WW1 继续 ON。

节拍 2：在节拍 1，MM1 ON，进入此节拍。这时，由于 MM1 ON，而使 WW1 OFF。

节拍3：在节拍2，刀架转动到了挡块位置，XK1 又压下、ON，进入此节拍。这时，WW2 ON，显示位置2。而 MM1 继续 ON。

节拍4：在节拍3，WW2 ON，进入此节拍。这时，由于 WW2 ON，而使 MM1 OFF。

节拍5：在节拍4，刀架转动，XK1 松开、OFF，进入此节拍。这时，MM2 ON，而 WW2 继续 ON。

节拍6：在节拍5，MM2 ON，进入此节拍。这时，由于 MM2 ON，而使 WW2 OFF。

节拍7：在节拍6，刀架转动到了挡块位置，XK1 又压下、ON，进入此节拍。这时，WW3 ON，显示位置3。而 MM2 继续 ON。

节拍8：在节拍7，WW3 ON，进入此节拍。这时，由于 WW3 ON，而使 MM2 OFF。

节拍9：在节拍8，刀架转动，XK1 松开、OFF，进入此节拍。这时，MM3 ON，而 WW3 继续 ON。

节拍10：在节拍9，MM3 ON，进入此节拍。这时，由于 MM3 ON，而使 WW3 OFF。

节拍11：在节拍10，刀架转动到了挡块位置，XK1 又压下、ON，进入此节拍。这时，WW1 ON，显示位置1。而 MM3 继续 ON。

节拍12：在节拍11，WW1 ON，进入此节拍。这时，由于 WW1 ON，而使 MM3 OFF。逻辑复原，与状态0相同。

以上分析结果记录见表3-9。

表 3-9 梯形图通电表

节拍	输入	内部状态				输出		
	当前输入	XK1	MM1	MM2	MM3	WW1	WW2	WW3
0		1	0	0	0	1	0	0
1	$\overline{XK1}$	0	1	0	0	1	0	0
2	MM1	0	1	0	0	0	0	0
3	XK1	1	1	0	0	0	1	0
4	WW2	0	0	0	0	0	1	0
5	$\overline{XK1}$	0	0	1	0	0	1	0
6	MM2	0	0	1	0	0	0	0
7	XK1	1	0	1	0	0	0	1
8	WW3	0	0	0	0	0	0	1
9	$\overline{XK1}$	0	0	0	1	0	0	1
10	MM3	0	0	0	1	0	0	0
11	XK1	1	0	0	1	0	0	0
12	WW1	1	0	0	0	1	0	0

显然，每当 XK1 从 OFF 到 ON，说明刀架转过了一个位置。而从表中可知，这里相应的显示也将在 WW1 到 WW3 之间依次变换。因而，它正是有了这个时序逻辑，仅用一个行程开关，也可显示刀架的实际位置。当然，这里在原始位置时，须使 WW1 及 MM3 初始化为 ON。

2. 分析图 3-29 时序逻辑

它的逻辑表达式为

$$WW1 = XK1 * \overline{MM1} + (\overline{XK1} + \overline{MM1}) * WW1$$

$$MM1 + \overline{XK1} * WW1 + (XK1 + WW1) * MM1$$

这里主令器件也是 XK1，所有节拍的转换都由它的变化引起。原始状态时 XK1 不输入，为 0，内部器件 MM1、输出器件 WW1 均 OFF。

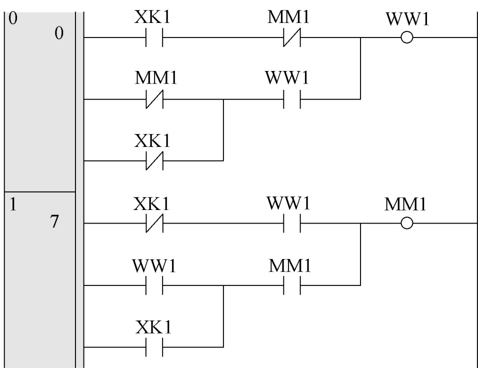


图 3-29　单按钮起停电路

节拍 0：XK1 松开、OFF，MM1 OFF，WW1 OFF。

节拍 1：在节拍 0，XK1 按下、ON，进入此节拍。这时，WW1 ON，MM1 继续 OFF。

节拍 2：在节拍 1，XK1 松开、OFF，进入此节拍。这时，MM1 ON，而 WW1 继续 ON。

节拍 3：在节拍 2，XK1 又压下、ON，进入此节拍。这时，WW1 OFF，而 MM1 继续 ON。

节拍 4：在节拍 3，XK1 松开、OFF，进入此节拍。这时，MM1 OFF，而 WW1 继续 OFF。逻辑复原。

其结果见表 3-10。

表 3-10　图 3-29 电路通电表

节拍序号	当前输入	输入器件 XK1	内部器件 MM1	输出器件 WW1
0		0	0	0
1	XK1	1	0	1
2	$\overline{XK1}$	0	1	1
3	XK1	1	1	0
4	$\overline{XK1}$	0	0	0

从表中可知，它仅用一个按钮，第 1 次按下，WW1 ON，起动。而第 2 次再按下，WW1 OFF，则停止。这里的又是起、保、停逻辑。

3. 分析图 3-1 梯形图程序

它为随机电路，故要把它所有的可能情况都列出（如能看得清楚，“对称”的情况也可不列）。表 3-11 为图 3-1 梯形图的通电表。表只列了逆时针的情况。顺时针的与这个对称，就不画了。

从分析知，它是能实现所述的功能的，即，不管手柄处于哪个位置，按下 A1 后，应使其转到使 XK1 合下的位置；按 A2，则到使 XK2 合下的位置；按 A3，则到使 XK3 合下的位置。

3.3.3　异步时序逻辑综合

综合是分析的反问题。组合逻辑综合，由于没有反馈，关系简单些，但也不易入手。而异步时序逻辑，还有反馈，还要用到一些内部器件，去记录一些节拍的状态。怎么入手，就更难了。

表 3-11 图 3-1 梯形图通电表

情况	节拍	当前输入	输入元件情况						中间及输出器件情况		
			A1 0.01	A2 0.02	A3 0.03	XK1 1.01	XK2 1.02	XK3 1.03	10.02	CT1 5.01	CT2 5.02
1	0		0	0	0	0	0	1	0	0	0
	1	A1	0	1	0	0	0	1	1	1	0
	2	$\overline{A1}$	0	0	0	0	0	1	1	1	0
	3	$\overline{XK3}$	0	0	0	0	0	0	1	1	0
	4	XK2	0	0	0	0	1	0	0	0	0
2	…	…	…								
	0		0	0	0	0	0	1	0	0	0
	1	A2	1	0	0	0	0	1	0	1	0
	2	$\overline{A2}$	0	0	0	0	0	0	0	1	0
	3	$\overline{XK3}$	0	0	0	0	0	0	0	1	0
	4	XK2	1	0	0	0	1	0	0	1	0
	5	$\overline{XK2}$	0	0	0	0	0	0	0	1	0
	6	XK1	1	0	0	1	0	0	0	0	0
3	…	…									
	0		0	0	0	0	1	0	0	0	0
	1	A3	1	0	0	0	1	0	0	1	0
	2	$\overline{A3}$	0	0	0	0	1	0	0	1	0
	3	$\overline{XK2}$	0	0	0	0	0	0	0	1	0
	4	XK1	1	0	0	1	0	0	0	0	0

此外，综合是根据要求实现的功能去设计程序。同样的要求，实现的程序可能很多。所以，综合还有个优化问题。要在很多可实现的方案中，优选一个最优的方案。这也加大了综合的难度。

异步时序逻辑综合方法很多，但是如用通电表作为设计起点，则比较方便。其方法是按要求确定原始通电表。进而设计通电表。接着，根据通电表，列写输出及内部继电器的逻辑表达式。最后，再化简表达式、画出梯形图。

异步时序逻辑综合目标也就是 PLC 编程要实现的目标。只是这里着重从逻辑上考虑。那么，怎么用通电表作为设计的起点呢？这还要从电路正常工作应遵守的原则谈起。

1. 电路正常工作应遵循的原则

电路，触点电路也好，梯形图这样异步时序逻辑也好，其正常工作要遵守惟一性原则。

（1）触点电路。某种逻辑条件，即有关的输入元件（输入继电器）及内部辅助继电器、输出继电器的一组状态组合，对应的输出取值是惟一的。要想用相同的逻辑条件产生不同的输出，是不可能的。

如用一个钮子开关控制一个指示灯，若钮子扳把有甲、乙两个位置，可以有这样的情况，钮子扳把处在甲位置时灯亮，处在乙位置时灯不亮；但不能有这样的情况，同样处于甲

位置，灯有时亮，有时不亮。可以想象，除了另有所控（还有别的逻辑条件），否则这是不可能的。

从本质上讲，这是因为逻辑条件与逻辑输出之间的关系为单值函数，一种输入只对应一种输出。违背这个原则设计的触点电路，逻辑关系是混乱的，其设计意图也是不可能实现的。一般称这种情况为逻辑条件相混，简称相混。

（2）梯形图逻辑。它输出与输入之间关系，不是惟一对应的。这里主要是输出点（位）、内部器件（位）都有“记忆”的作用，可用本身触点反馈；也可用置位（S）指令实现这个“记忆”。有了这个“记忆”可实现同一输入，可产生不同的输出。

在通电表中，如果输出点（位）、内部器件（位）有多个连续为 1、ON 的节拍，则把其中所有第一个 1、ON 的节拍，定义为起动节拍，其相应的动作称起动；连续 1、ON 后的第一个 0、OFF 节拍，定义为断电节拍，其相应的动作称断电。

有了这个定义，梯形图电路的惟一性原则可表述为：在某种逻辑条件下，所对应的输出点（位）、内部器件（位）的起动、断电应是惟一的。要想在相同的逻辑条件下，使输出点（位）、内部器件（位）在某个节拍起动（或断电），而在另一个节拍又不起动（或断电）是不可能的。

这是因为，异步时序逻辑“分解”之后，起动与保持电路分别也都是组合逻辑，也是单值的，因而也应遵循这个原则。

所设计的通电表如不满足这个惟一性原则，也称逻辑条件相混。可增加内部器件（位）以增加逻辑条件的变量，避免相混。

（3）实例说明。设计一个时序电路，要求是用输入点 XK1 控制输出 WW1，各节拍的对应关系见表 3-12。到第 4 节拍时，电路又复原，回到原始状态。

表 3-12　各节拍输入、输出对应关系

节拍序号	当前输入	输入器件 XK1	输出器件 WW1
0		0	0
1	XK1	1	1
2	$\overline{\text{XK1}}$	0	1
3	XK1	1	0
4	$\overline{\text{XK1}}$	0	0

从表可知，这里的 1、3 两个节拍，逻辑条件相同，但 1 节拍要求 WW1 起动，而 3 节拍又要求其断电，这种矛盾的要求，若没有别的措施（即再增加内部器件）是无法实现的。从断电检查，1、3 节拍也有类似的情况。

为了排除这个相混，则要加一个内部继电器 MM1。如其通、断电过程见上表 3-10 所示。由于这时 WW1、MM1 可互为条件，所以，无论对 MM1，或 WW1，它的通断电都不相混，因而这个通电表也就不相混了。

2. 梯形图逻辑综合算法

惟一性原则给梯形图工作设置了约束，但也给梯形图设计提供了算法。其步骤是：

（1）初列通电表。根据设计要求，按输出时序划分工作步，确定各个步的输入与输出的对应关系，初列通电表。这个表也称原始通电表。

这个设计不难。因为它仅是设计要求的“表格化”。

（2）惟一性设计。对原始通电表进行惟一性检查，如不满足惟一性原则，可适当增加内部继电器，以得到一个合乎惟一性原则的通电表。

具体作法是：

1）惟一性检查。依次对在原始通电表中的，每一输出继电器的起动与断电进行惟一性检查；

把所有起动的步或节拍的逻辑条件与所有不工作（OFF）节拍的逻辑条件进行比较，看是否存在相混；

把所有断电的步或节拍的逻辑条件与所有工作（ON）节拍的逻辑条件进行比较，看是否存在相混；

若有，建相混表。这个表可附在原始通电表右侧，亦即原始通电表的扩展。将逻辑条件相同者分成一组，占一列，用英文字母命名。在起动（或断电）节拍，标大写字母。在其它与逻辑条件相混的节拍，标与其同名的小写字母。然后，在同名的大小写字母间的节拍画上竖实线，其余部分画上竖虚线。画时，应把第一节拍看做最后一个节拍的次拍。这主要因为电路总是要循环工作的。表 3-13 所示的就是对表 3-12 检查后建的相混表。

表 3-13　经检查后所建的相混表

节拍序号	当前输入	输入器件 A(00000)	输出器件 01000		
0		0	0		
1	A	1	1	A	b
2	$\overline{A}$	0	1		
3	A	1	0	a	B
4	$\overline{A}$	0	0		

2）增加内部继电器。在各组的实线处须建立分界线，以便避免相混。办法是增加内部继电器。新增的内部继电器在分界线之前为 OFF，而之后为 ON，把它作为因子，加入逻辑条件中，显然，这一样的相混将不再有了。

新增的内部继电器从 OFF 到 ON 之后，在循环结束之前要回到 OFF，以保证电路能周而复始地循环工作。这个从 ON 到 OFF 可安排在最后一个节拍。

此外，对新增的内部继电器的起动与断电逻辑条件也要作相混检查。才能保证它能正常工作。检查办法同前。

如新增的内部继电器也存在相混，则也要再建分界线。以至于还要再查，再建，直到不再有相混为止。

提示：也可在通电表惟一性设计之前，先检查输入信号的情况。如有短信号，则要用内部继电器对其“记忆”，建分界线；如有计数信号，则要对它的每个输入节拍都要：“记忆”，建分界线。读者可用这里介绍的办法检查，证明这个结论是正确的。

3）列写逻辑式子。分起动电路及保持电路列写逻辑式子。

（a）起动电路逻辑表达式

要求是，起动节拍的逻辑条件应为 1，无电节拍应为 0，其它节拍可任意。表达式可分为特解和一般解。特解仅对起动节拍进行分析，求其为 1 时的逻辑条件；一般解还把无关项

（可任意取值的项）也考虑进去。特解为

$$J_{QT} = \sum_{1}^{N} A_{Qi} \cdot M_{Qi}$$

式中　N——起动次数；

A_{Qi}——起动节拍，当前输入；

M_{Qi}——起动节拍的其它元件状态组合。

一般解为

$$J_{QY} = \overline{\sum_{1}^{Q} A_{WR} \cdot M_{WR}}$$

式中　Q——所有 OFF 节拍的总数；

$A_{WR} \cdot M_{WR}$——OFF 节拍的逻辑条件。

一般讲，特解含的因子多，所用的触点多，一般解简单，但也可能包含多余的项。正确的选择是化简一般解，并从中挑选出含特解的项。但要确保起动节拍的当前输入成为必备因子。

（b）保持电路逻辑表达式

要求是：断电节拍的逻辑条件为 0（对锁存指令的 R 电路，则为 1），ON 节拍为 1（R 电路，为 0），其它的可以任意取值。也分为特解与一般解。特解仅考虑 ON 节拍的条件，一般解还考虑任意项。特解式为

$$J_{BT} = \left(\sum_{1}^{r} A_{rj} \cdot M_{rj}\right) J$$

式中　r——所有为 ON 的节拍数；

$A_{rj} \cdot M_{rj}$——为 ON 的节拍的逻辑条件；

J——求解元件自身触点，即自保触点。

一般解为

$$J_{BY} = \left(\overline{\sum_{1}^{N} A_{Di} \cdot M_{Di}}\right) J$$

式中　N——断电次数；

A_{Di}——断电节拍的当前输入；

M_{Di}——断电节拍的其它元件状态组合；

J——求解元件自身触点，即自保触点。

化简时，同样也是先化简一般解，然后选含特解的项。但要确保断电节拍的当前输入的反成为必备因子。

在使用锁存指令时，其 S 电路的特解及一般解，与起动电路的有关逻辑表达式相同。其 R 电路表达式为，除去 J 后的非，并将特解与一般解互换。即特解为

$$R_{T} = \sum_{1}^{N} A_{Di} \cdot M_{Di}$$

一般解为

$$R_{Y} = \overline{\sum_{1}^{p} A_{rj} \cdot M_{rj}}$$

式中右边各项含义与保持电路表达式中的各项含义相同。

（3）逻辑化简、列写指令、画梯形图。梯形电路化简可分两组进行：起动电路（或 S 电路）及保持电路（或 R 电路）。用的是多输出触点电路化简的方法。

要注意的是，它有不少不使用状态，可把它作为无关项（可为 1，也可为 0）。如 Bi，只要电路未起动，它为 1、为 0 都无关紧要；Qi，已起动工作之后，它为 1、为 0 也无关紧要。无关项选取合适的值，用以化简，可使 Qi、Bi 更大地简化。

对逻辑式子进行化简是数字逻辑问题。这里不再赘述。

画梯形图：多数 PLC 厂家都提供有梯形图编程软件，故可依逻辑式子与梯形图的对应关系，直接画梯形图。

3.3.4　异步时序逻辑设计实例

1. 液体混合罐工作控制

有一两种液体混合装置，如图 3-30 所示。控制要求是，起始状态容器是空的，3 个阀门（X1、X2、X3）均关闭，电动机 M 也不工作。液面传感器 L、I、H 也处于 OFF 状态。

起动后，先是 X1 阀门打开，液体 A 流入。液位达到 I 时，I 开关 ON，使 X1 阀门关闭，而 X2 打开，液体 B 流入。当液面到达 H 时，H 开关 ON，X2 阀门关闭，起动电动机 M，对两种液体作搅拌。搅拌 6s 后，电动机 M 停止工作，打开阀门 X3，把混合液放出，直到 L 传感器 OFF，再过 2s，阀门 X3 关闭，又开始新的周期。若要停止操作，可按停车按钮 T。但按后不立即停止工作，待这个工作循环完成，才停止工作。

图 3-30　两种液体进行混合的装置

设计过程

1）用的可编程序控制器可任选，其 I/O 分配略，但用相应符号 X1、X2、X3、Q、T、M、H、L 代表。时间继电器符号为 TM、TL。

2）用梯形图逻辑编程。其步骤是列原始通电表，通电表惟一性设计，列写逻辑式子，画梯形图。

列原始通电表：用相应的符号列表，如起动按钮用 Q，TM 为与 M 共同工作的时间继电器等。所画出的原始通电表见表 3-14。

表 3-14　图 3-30 原始通电表

节拍	当前输入	H	I	L	Q	T	X1	X2	X3	M	TM	TL
0		0	0	0	0	0	0	0	0	0	0	0
1	Q	0	0	0	1	0	1	0	0	0	0	0
2	$\overline{Q}$	0	0	0	0	0	1	0	0	0	0	0
3	L	0	0	1	0	0	1	0	0	0	0	0
4	I	0	1	1	0	0	0	1	0	0	0	0
5	H	1	1	1	0	0	0	0	0	1	1	0
6	TM	1	1	1	0	0	0	0	1	0	0	0
7	$\overline{H}$	0	1	1	0	0	0	0	1	0	0	0
8	$\overline{I}$	0	0	1	0	0	0	0	1	0	0	0
9	$\overline{L}$	0	0	0	0	0	0	0	1	0	0	1
10	$\overline{TL}$	0	0	0	0	0	0	0	0	0	0	0

对原始通电表进行惟一性检查，知：

X1 起动主要靠 Q 信号，其它 X1 OFF 的节拍均无此信号，所以，不存在相混。但是，第二循环及以后的循环，无 Q 信号，仍应使 X1 起动，这可用 TL 帮忙。这相当于把 1、10 节拍合并。X1 断电，其信号为 I，其它 ON 节拍也无此信号，故也不存在相混。

X2 于第 4 节拍工作，其它节拍都不工作。第 4 节拍时 I、L 均 ON H OFF。这种情况还出现在第 7 节拍。但第 7 节拍时 X3 ON，而第 4 节拍时 X3 OFF，这可把第 4 与第 7 节拍的逻辑条件区分开。故对 X2 而言，惟一性原则也满足。

X3 于第 6 节拍起动，它用的信号为 TM，是惟一的。其断电于第 10 节拍，用的信号为 TL 也是惟一的。

M 于第 5 节拍工作，这时 H ON。第 6 节拍也是这个情况。但两者可用 TM 区分开，故 M 也不存在相混。

TM 靠 H　ON 起动，是惟一的。

TL 靠 X3 ON，再 L OFF 起动，也是惟一的。

这样，通电表的惟一性设计后，可保持原始通电表不变。

停车按钮 T 输入是随机的，但它输入后可对其进行记忆（如以 Z 表示），并用这记忆的信号去“切断”TL 与 X1 的连系，即可达到目的。其在通电表中表示略。

列写逻辑式子：

为了便于理解及使式子简练，这里用的也是原始符号。

对 X1：其起动电路，依上述分析应为 $Q+\bar{Z}TL$；其保持电路应为 $\bar{I}X1$。

对 X2：其保持电路不用，起动电路用作工作电路，应为 $I\,\bar{H}\,\overline{X3}$。

对 X3：其起动电路为 TM；其保持电路为 $\overline{TL}X3$。

对 M：其保持电路不用，起动电路即为工作电路，为 $H\,\overline{TM}$。

对 TM：工作电路为 H。

对 TL：工作电路为 $X3\bar{L}$。

对 Z：其起动电路为 T；保持为 $\overline{TL}Z$。

这样，它的完整的逻辑式子为

$$X1 = Q+\bar{Z}TL+\bar{I}X1$$
$$X2 = I\,\bar{H}\,\overline{X3}$$
$$X3 = TM+\overline{TL}X3$$
$$M = H\,\overline{TM}$$
$$TM = H$$
$$TL = X3\bar{L}$$
$$Z = T+\overline{TL}Z$$

根据以上逻辑式画梯形图，如图 3-31 所示。

这里列写逻辑式子未完全套用以前的公式，而是用直接观察的办法。由于本例逻辑变量较多，相区分的信号又较明显，直接观察更为简便。

图 3-32 为利用了锁存指令的梯形图。其功能与图 3-31 相同。

2. 小车运动控制

如图 3-33 所示的小车，有 3 个状态，向左（反转），向右（正转），停车。LS 为反映小

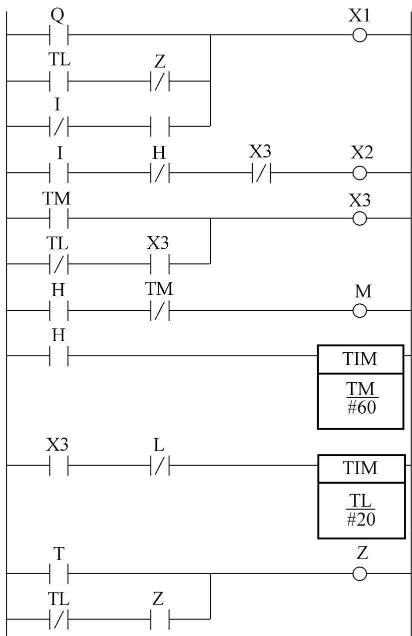


图 3-31 液体混合罐工作控制程序

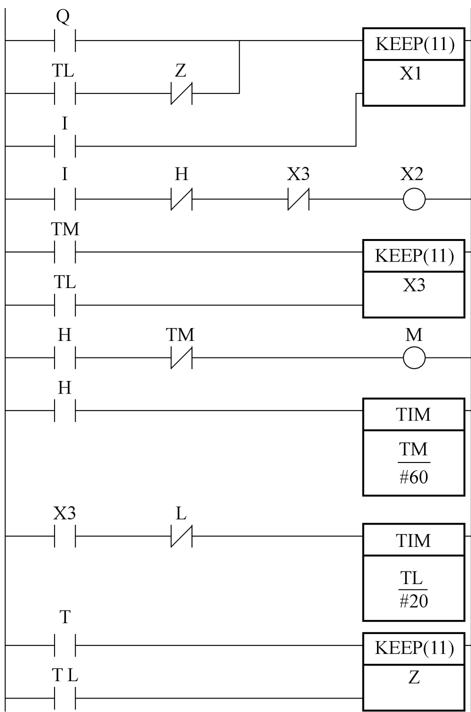


图 3-32 使用锁存指令梯形图

车所处位置的行程开关，PS 为选择小车位置的按钮，各有 5 个。控制要求是：按下选择按钮的编号大于小车当前位置的开关号时，按下起动按钮 SW，小车将向右运动，直至小车新位置行程开关的编号与选择位置的开关编号相等时，小车停止运动。按下选择按钮的编号小于小车当前位置行程开关的编号时，按下起动按钮 SW，小车将向左运动，直至小车新前位置行程开关的编号与选择位置的开关编号相等时，小车停止运动。

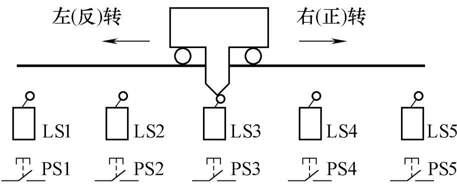


图 3-33 小车运动控制示意图

设计过程

1）输入用符号 LS1、LS2、LS3、LS4、LS5、PS1、PS2、PS3、PS4、PS5、SW 代表，分别对应的输入点好为 0.01、0.02、…、0.10，而起动按钮（SW）对应点号为 0.00。输出用符号 10.01 代表向右（正转）、10.02 代表向左（反转）。

2）编程

（a）通电表设计

所设计的电路为随机电路，从输入入手，按所有可能情况列写通电表，相当复杂，也无此必要。如果从输出考虑，由于它只有向左、向右两种情况，故便于归纳。先不考虑起动按钮，仅考虑行程开关 LS 及选择按钮 PS 与向右、向左输出的置位与复位的逻辑关系，其通电表见表 3-15。由于它是随机的，故这里省略了节拍的概念。节拍是与输入相联系的概念，现从输出考虑，故可不用它。

表 3-15　图 3-33 通电表

<table>
<tr><th colspan="2">输出</th><th colspan="10">输　入</th></tr>
<tr><th>10.01</th><th>10.02</th><th>LS1</th><th>LS2</th><th>LS3</th><th>LS4</th><th>LS5</th><th>PS1</th><th>PS2</th><th>PS3</th><th>PS4</th><th>PS5</th></tr>
<tr><td rowspan="10">S</td><td rowspan="10"></td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td rowspan="4">0</td><td rowspan="4">0</td><td rowspan="4">0</td><td rowspan="4">0</td><td rowspan="4">1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td rowspan="3">0</td><td rowspan="3">0</td><td rowspan="3">0</td><td rowspan="3">1</td><td rowspan="3">0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2">0</td><td rowspan="2">0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>R</td><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td rowspan="3">R</td><td rowspan="3">R</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td></td><td>R</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td rowspan="10"></td><td rowspan="10">S</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td rowspan="4">1</td><td rowspan="4">0</td><td rowspan="4">0</td><td rowspan="4">0</td><td rowspan="4">0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td rowspan="3">0</td><td rowspan="3">1</td><td rowspan="3">0</td><td rowspan="3">0</td><td rowspan="3">0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2"></td><td rowspan="2">0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td rowspan="5"></td><td rowspan="5"></td><td rowspan="5">0</td><td rowspan="5">0</td><td rowspan="5">0</td><td rowspan="5">0</td><td rowspan="5">0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td></td><td></td><td colspan="5">任意组合</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

从表知，对 10.01、10.02，其 S、R 电路的逻辑条件，与可能出现的逻辑条件均不相混，故此表可满足惟一性原则。

（b）列写逻辑式

由于这里的输入 LS、PS 出现时，都仅为一个 ON，这为我们列写逻辑式提供方便，即

仅考虑变量本身，其它可不考虑。具体列写如下：

10.01S 电路

$$S1 = PS5(LS4 + LS3 + LS2 + LS1) + PS4(LS3 + LS2 + LS1) + PS2(LS2 + LS1) + PS2LS3$$

10.01R 电路

$$R1 = PS5LS5 + PS4LS4 + PS3LS3 + PS2LS2$$

10.02S 电路

$$S2 = PS1(LS2 + LS3 + LS4 + LS5) + PS2(LS3 + LS4 + LS5) + PS3(LS4 + LS5) + PS4LS5$$

10.02R 电路

$$R2 = PS1LS1 + PS2LS2 + PS3LS3 + PS4LS4$$

（c）画梯形图

列出逻辑式后，可画出对应的梯形图，不过还要考虑几个实际问题：

① 选择按钮给出的是短信号，按后即复原，故须对其记忆。设用内部辅助继电器200.01～200.05分别对PS1～PS5作记忆，直到选择编号与实际编号相等时，再清除这个记忆。以10.01为例，其逻辑式为

$$200.01 = PS1 + \overline{LS1}(200.01)$$

其余的类推。这里用到位后的信号LS1作为断电输入信号，故其保持电路为$\overline{LS1}$(200.01)。

用了200.01～200.05后，即可用它代换上述逻辑式中的PS1～PS5。

② 起动输入信号SW应作为10.01、10.02的置位电路的条件之一。

③ 10.02、10.01必须互锁，以保证安全。

④ 实际电路应尽可能简化，以节省指令条数。

考虑以上4点后的梯形图如图3-34所示。图中10.02的输出未画出，它与10.01类似。

本梯形图把KEEP 10.01等指令画在前而OUT 200.01等在后，这很重要。这可保证到达要求位置时，10.01等先复位，然后200.01等才复位。否则，即前后调一下，10.01等就复位不了的。

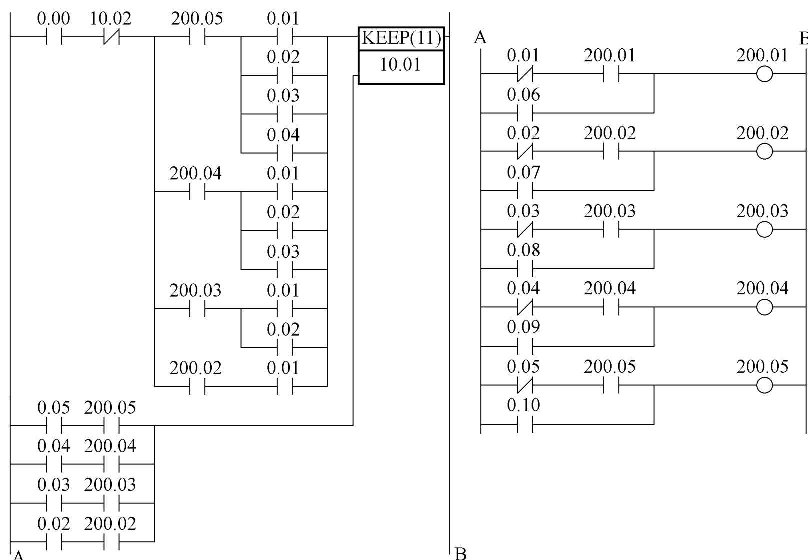


图3-34　小车运动控制程序

3. 组合机床动力头运动控制

设计组合机床主轴动力头的运动控制的电路。该动力头由液压驱动。电磁阀 DT1 得电，前进；失电后退。同时，还用电磁阀 DT2 控制前进及后退的速度，得电快速，失电慢速。

要求机床的工作过程是：从原位（行程开关 1XK ON）开始工作。按下起动按钮 Q，先快速进；到行程开关 2XK ON，转为工进（慢速前进）；加工一定深度，3XK ON，快退；退到 2XK OFF（目的为了排屑），又快进；快进至 3XK ON，又转工进；加工到尺寸，4XK ON，快退，直至原位 1XK ON 停，完成一个工作循环。

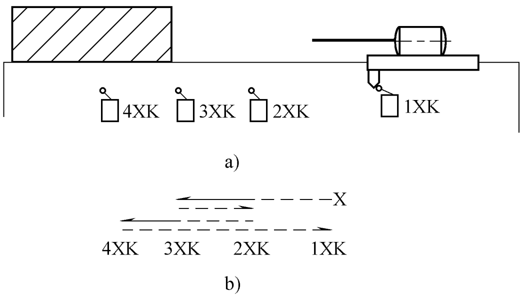


图 3-35　机床动力头运动控制

图 3-35a 所示为系统结构。其具体工作过程如图 3-35b 所示。这里，虚线为快速运动，而实线为慢速运动，箭头指明它的运动方向。

设计过程

1）输入输出分配

用 CPM2A CPU20 的可编程序控制器，其点数已够用。I/O 分配如下：

输入：Q、1XK、2XK、3XK、4XK 分别为 0.00、0.01、0.02、0.03、0.04。

输出：DT1、DT2 分别为 10.01、10.02。

2）程序设计

（a）通电表设计。先依照工作顺序，初列通电表，见表 3-16。

（b）对原始通电表进行惟一性原则检查。检查 DT1。它有两次起动。第一次起动输入信号为 Q，用它可与所有 DT1 为 OFF 节拍的逻辑条件区分开。第二次起动在第 7 节拍。其起动信号为 $\overline{2XK}$ 及其它条件均与第 13 节拍（DT1 为 OFF）相混，即表 3-16 中标的 B—b。DT1 有两次断电，第一次在第 5 节拍，它与第 9 节拍的条件相混，即表 3-16 中标的 A—a。第二次断电逻辑条件不相混。

检查 DT2。它有 3 次起动，第一次起动不存在相混。第二次在第 5 节拍起动，与第 9 节拍相混，即表示所标 A—a。第三次在第 10 节拍起动，不存在相混。它也有三次断电，第一次在第 4 节拍断电，它与第 8 节拍相混，即表中标以 C—c。第二次在第 9 节拍断电，如果作到在第 5 节拍 DT1 断电后，DT2 再起动，可不相混。第三次断电在第 14 节拍，不存在相混。

按规定画完实线后（见表 3-16），须在第 3（或 1，2）、6 及 10（或 11、12）节拍建 3 条分界线，才能把全部相混区分开。再查所建的分界线也不存在相混。

3 条分界线，需用三个内部辅助继电器。设用 9.01 及 9.02。其工作情况也按“先逐个 ON，ON 后再逐个 OFF”的原则布置，如表 3-16 所示。

表 3-16　通电表设计

节拍	当前输入	Q	1XK	2XK	3XK	4XK	DT1	DT2	扩展表				9.01	9.02
0		0	1	0	0	0	0	0					0	0
1	Q	1	1	0	0	0	1	1					0	0

（续）

节拍	当前输入	Q	1XK	2XK	3XK	4XK	DT1	DT2	扩展表				9.01	9.02
2	$\overline{Q}$	0	1	0	0	0	1	1					0	0
3	$\overline{1XK}$	0	0	0	0	0	1	1					1	0
4	2XK	0	0	1	0	0	1	0			C		1	0
5	3XK	0	0	1	1	0	0	1	A				1	0
6	$\overline{3XK}$	0	0	1	0	0	0	1					1	1
7	$\overline{2XK}$	0	0	0	0	0	1	1		B			1	1
8	2XK	0	0	1	0	0	1	1			c		1	1
9	3XK	0	0	1	1	0	1	0	a				1	1
10	4XK	0	0	1	1	1	0	1					0	0
11	$\overline{4XK}$	0	0	1	1	0	0	1					0	0
12	$\overline{3XK}$	0	0	1	0	0	0	1					0	0
13	$\overline{2XK}$	0	0	0	0	0	0	1		b			0	0
13	1XK	0	1	0	0	0	0	0					0	0

对表 3-16 再做检查，可知无论是 DT1、DT2，还是对 10.01、10.02 均不相混。

（c）列写逻辑式子

DT1 起动电路特解

$$Q_{T1}=Q\cdot 1XK\cdot \overline{2XK}\cdot \overline{3XK}\cdot \overline{4XK}\cdot \overline{(9.01)}\cdot \overline{(9.02)}+\overline{Q}\cdot \overline{1XK}\cdot \overline{2XK}\cdot \overline{3XK}\cdot \overline{4XK}\cdot DT2\cdot (9.01)\cdot (9.02)$$

DT1 起动电路通解，经化简后为

$$Q_{y1}=Q+\overline{2XK}\cdot (9.01)+\overline{2XK}\cdot (9.02)+4XK\cdot (9.01)+\cdots$$

从中选出含有特解的项。可以是头两项，或第一及第三项。这里用头两项，即

$$Q_1=Q+\overline{2XK}\cdot (9.01)$$

DT1 保持电路特解为第 1、2、3、4 及 7、8、9 节拍逻辑条件的或，其表达式略。

DT1 保持电路通解为第 5 及第 10 节拍逻辑条件或的非，经化简为

$$B_{y1}=[Q+1XK+\overline{2XK}+\overline{3XK}+\overline{4XK}\cdot \overline{(9.01)}+\overline{4XK}\cdot (9.02)]\cdot DT1$$

选$\overline{3XK}$及$\overline{4XK}\cdot(9.02)$ 或$\overline{4XK\ (9.01)}$即可覆盖所有的特解，现选$\overline{4XK}$（9.02）。这样可得

$$DT3=Q+\overline{2XK}(9.01)+(\overline{3XK}+\overline{4XK}\cdot (9.02)\cdot DT1)$$

同理，可求 DT2、9.01、9.02 的逻辑表达式：

$$DT2=Q+3XK\cdot (9.01)\cdot \overline{(9.02)}+4XK+(\overline{2XK}+\overline{(9.01)}+(9.02)+DT1)\cdot (\overline{3XK}+\overline{(9.02)}+\overline{(9.01)}+\overline{DT1})\cdot (\overline{1XK}+DT1)\cdot DT2$$

$$9.01=\overline{1XK}\cdot DT1\cdot \overline{2XK}+\overline{4XK}\cdot (9.01)$$

$$9.02=\overline{3XK}\cdot \overline{DT1}\cdot (9.01)+\overline{4XK}\cdot (9.02)$$

（d）画梯形图。依上述逻辑表达式，可画出的对应梯形图，如图 3-36 所示。

若用锁存指令（KEEP），那起动电路即为 S（置位）电路，保持电路（去掉自保持触点）的反即为 R 电路。其梯形图略。

4. 计数信号应用

（1）设计要求。要求设计用一个按钮控制的电路，按钮 A 反复动作，到第 4 节拍时，产生输出，100.00 ON。第 6 节拍时，停止输出，10.00 OFF，电路复原。具体的要求见原始通电表，即表 3-17。

（2）惟一性设计。从起动看，第 4 节拍和第 2 及第 6 节拍相混；从断电看，第 6 节拍和第 4 节拍相混。按上述原则，画出它的相混表，见表 3-18。

可看出，只要在 3、5 节拍处建分界线，即可把上述相混区分开。再查所建分界线是否相混时，发现第 3 节拍的分界线建立条件与第 1 节拍相混，故应在第 2 节拍再建一分界线。再查第 2 节拍的分界线又与第 0 节拍（即第 6 节拍）相混，故还应于第一节拍建分界线。之后再查，再不相混了。

从表 3-18 知，须建 4 条分界线，须用 2 个内部辅助继电器，并设其为 9.01、9.02。

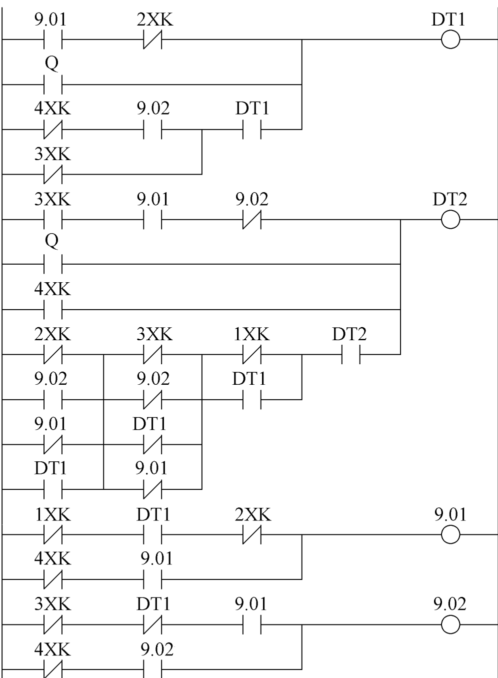


图 3-36　组合机床动力头运动控制程序

表 3-17　原始通电表

节　　拍	当 前 输 入	A	10.00
0		0	0
1	A	1	0
2	$\overline{A}$	0	0
3	A	1	0
4	$\overline{A}$	0	1
5	A	1	1
6	$\overline{A}$	0	0

表 3-18　通电表设计

节拍	当前输入	A	10.00	
0		0	0	d　E　g
1	A	1	0	⋮　f
2	$\overline{A}$	0	0	d
3	A	1	0	F
4	$\overline{A}$	0	1	D　e
5	A	1	1	
6	$\overline{A}$	0	0	d　E

内部辅助继电器可按“依次通，全通后再依次断”的原则对其进行工作设定。如这里的 10.00 的取值为 0 的区，在第一条分界线处（即第一节拍）令 9.01 ON、9.02 OFF；第二条分界线处（即第 2 节拍），令 9.01 ON、9.02 ON（依次通的原则）；第三分界线处，令 9.01 OFF、9.02 ON（全通后，依次断个）。到第 5 节拍，则令 9.01 OFF、9.02 OFF，电路复原，又可建一个分界线。加入内部继电器 9.01 、9.02 后的通电表，见表 3-19。再查表，已无逻辑相混。

表 3-19 已无逻辑相混通电表

节 拍	当前输入	A	10.00	9.01	9.02
0		0	0	0	0
1	A	1	0	1	0
2	$\overline{A}$	0	0	1	1
3	A	1	0	0	1
4	$\overline{A}$	0	1	0	1
5	A	1	1	0	0
6	$\overline{A}$	0	0	0	0

（3）求逻辑表达式。有了表 3-19 所示通电表，下一步可求与其对应的逻辑表达式。

该表仅 4 个逻辑变量，用卡诺图求解较方便。卡诺图每格中有三种值：一为 1，是特解的最小项；一为 0，必须为 0 的最小项；一为 d，为任意项。分别对 10.00、9.01 、9.02 求解的卡诺图的步骤是：

1）填写卡诺图。每格都要依给定的逻辑条件填入合适的值。

2）画卡诺图。要把所有含 1 的格全覆盖。

3）列逻辑式。经选择把能包含所有 1，但又最简项组成逻辑表达式。

对 10.00、9.01 、9.02 求解的卡诺图，如图 3-37 所示。

（4）画梯形图。求出逻辑表达，其相应的梯形图如图 3-38 所示。

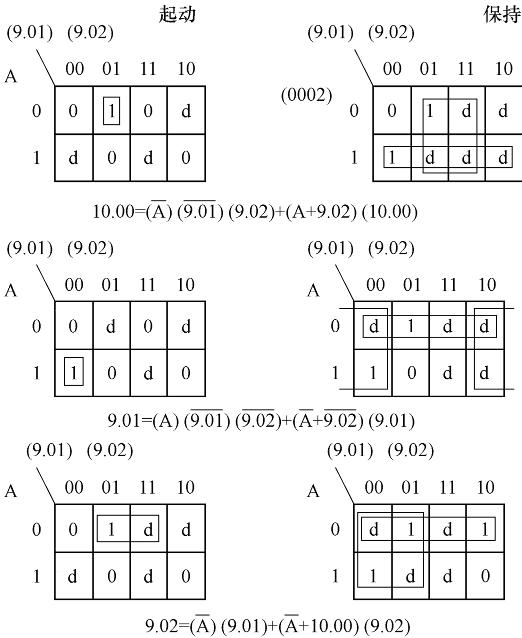


图 3-37 求解的卡诺图

5. 煤气加热炉切阀、转换阀控制

图 3-39 为煤气加热炉切阀、转换阀工作示意图。左右切阀、转换阀要定时切换，或按要求切换，以实现储热节能及保证切阀不超温。具体工作过程是，左、右煤气切阀 1 交替开通，空气、烟气转换阀左右转换，总是处于图 3-39a，或图 3-39b 状态之一。处于图 3-39a状态时，空气从左喷嘴进入炉子，并冷却左喷嘴（加温空气）；而烟气从右喷嘴排出，加温右喷嘴（储热）；左切阀进煤气，右切阀关闭。处于图 3-39b 态时，空气从右喷嘴进入

炉子，并冷却右喷嘴（加温空气）；而烟气从左喷嘴排出，加温左喷嘴（储热）；右切阀进煤气，左切阀关闭。

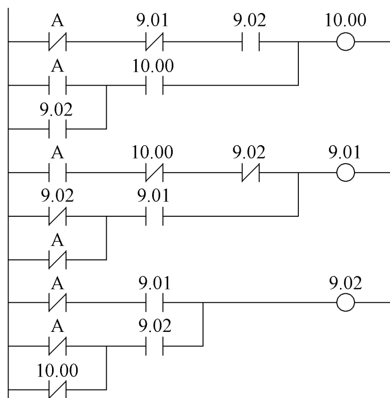


图 3-38 与求解卡诺图对应的梯形图程序

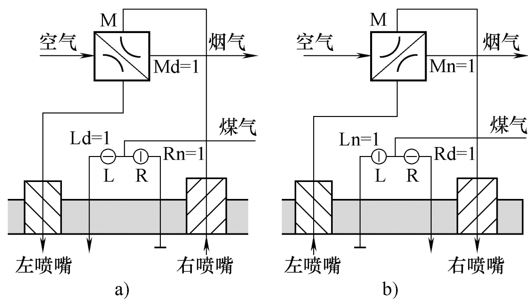


图 3-39 切阀、转换阀工作示意图

L—左切阀 R—右切阀 M—转换阀

输出用 3 个变量 L、M、R，分别代表左煤气切阀，空气、烟气转换阀，右煤气切阀。依以上描述，系统总是处于 100（左切阀开，转换阀置左位置）或 011（右切阀开，转换阀置右位置）状态之一。其切换的工艺要求是，每当送入控制脉冲 A，则先关已开的切阀，切阀关到位后，转换阀转向，换向到位后，再开原关闭的切阀。

阀动作反馈信号用相应的行程开关。具体有：左切阀开到位开关（Ld），左切阀关到位开关（Ln），右切阀开到位开关（Rd），右切阀关到位开关（Rn），转换阀置右位置开关（Md），转换阀置左位置开关（Mn）。

按设计要求，列出的通电表见表 3-20。表中有四处相混，但在第 4、第 8 及第 13 节拍加分界线即可消除这个相混。

为此，增加两个内部继电器 M01、M02。其状态在第 4、第 8 及第 13 节拍改变，见表3-20。

有了 M01、M02，此通电表已符合要求，已不存在相混。

表 3-20 图 3-39 通电表设计

节拍	当前输入	输入							输出				内部状态	
		Ld	Ln	Rd	Rn	Md	Mn	A	L	R	M		M01	M02
0		0	1	0	1	0	1	1	0	0	0		0	0
1	A	0	1	0	1	0	1	1	1	0	0		0	0
2	$\overline{A}$	0	1	0	1	0	1	0	1	0	0	c	0	0
3	Ld	1	0	0	1	0	1	0	1	0	0		0	0
4	A	1	0	0	1	0	1	1	0	0	0		1	0
5	$\overline{A}$	1	0	0	1	0	1	0	0	0	0		1	0
6	Ln	0	1	0	1	0	1	0	0	0	1	a C	1	0
7	Md	0	1	0	1	1	0	0	0	1	1	B d	1	0
8	Rd	0	1	1	0	1	0	0	0	1	1		1	1
9	A	0	1	1	0	1	0	1	0	0	1		1	1
10	$\overline{A}$	0	1	1	0	1	0	1	0	0	1		1	1
11	Rn	0	1	0	1	1	0	0	0	0	0	b D	1	1
12	Mn	0	1	0	1	0	1	0	1	0	0	A c	1	1
13	Ld	1	0	0	1	0	1	0	1	0	0		0	0
14	A	1	0	0	1	0	1	1	0	0	0		0	0

依此通电表，列出输出及内部继电器的逻辑表达式如下（化简过程略）：

$$M01 = A * Ld * \overline{M02} + (\overline{Ld} + \overline{M02}) * M01$$

$M02 = Rd + \overline{Ld} * M02$

$L = A * Ln * Rn * Mn + M01 * M02 * Mn + (Ln + \overline{A}) * L$

$R = Md * \overline{M02} + \overline{A} * R$

$M = Ln * M01 * \overline{M02} + (\overline{Rn} + \overline{M02}) * M$

依以上逻辑表达式，画出的梯形图如图 3-40 所示。

如图所示，在状态 100 时，送来脉冲 A，则先转换到 000（左切阀关）。左切阀关到位（Ln = 1），进入 010（转换阀置右位置）。阀置右到位（Md = 1），进入 011（右切阀开）。直到右切阀开到位（Rd = 1），这个周期完成。

同样，在状态 011 时，送来脉冲 A，则先转换到 010（右切阀关）。右切阀关到位（Rn = 1），进入 000（转换阀置左位置）。阀置左到位（Mn = 1），进入 100（左切阀开）。直到左切阀开到位（Rd = 1），这个周期也完成。

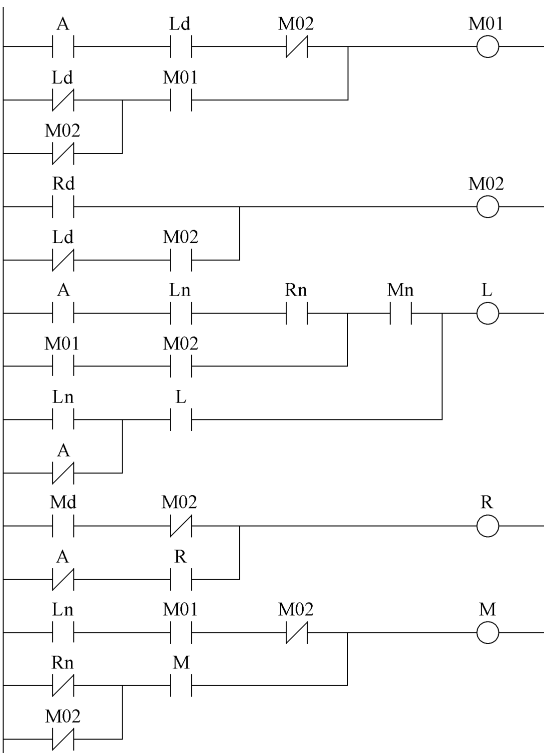


图 3-40 煤气加热炉切阀、转换阀控制程序

从图还可看到，如初始状态为 000，Ln = 1、Rn = 1、Mn = 1 时，如有 A 脉冲，则可转为 001。

3.4 同步时序逻辑编程

关键词：逻辑设计同步化、脉冲信号产生、前后逻辑条件一致、前后逻辑条件一致实现

3.4.1 同步时序逻辑表达式与状态图

1. 表达式

同步时序逻辑的表达式与异步时序逻辑的相同。也是分为起动与保持两个部分。也可用置位（S）及复位（R）指令处理。某个位的置位，其逻辑条件即为它的起动，与起动电路（Qi）对应，而复位则只是它的保持逻辑条件的非，与保持电路中 Bi 的非对应。为了方便，在同步时序逻辑中，置位（S）及复位（R）指令使用较多。

以刀架位置显示同步时序逻辑（见图 3-4）为例，列写其表达式如下：

这里的梯级 1、2 是把 XK1 信号变为脉冲信号，其表达式略。

梯级 3、4、5 为程序主体。同步时序逻辑的特点是，在脉冲信号作用期间，其先执行指令引起的输出变化，不会改变后执行指令的执行条件。故其输出表达式较简单，为

WW1 置位：$pA^{*}mW3$

WW1 复位：$pA^{*}mW1$

WW2 置位：$pA^{*}mW1$

WW2 复位：pA * mW2

WW3 置位：pA * mW2

WW3 复位：pA * mW3

梯级 6、7、8 为输出的记录。表达式更简单，这里略。

2. 状态图

状态图由一系列状态节点（圆圈及在其上的状态标识）及节点间的有向联线组成。用以代表同步时序逻辑各个节拍的状态及其转换与输出。

每一状态的标识都用一组布尔变量的不同取值表示。其各个位的取值，代表其因子（变量或变量的非）的乘积项（逻辑与）为 1。如某个状态，代表它的因子乘积项为 $x\bar{y}z$，那就用 101 表示它的状态。因为，只有当 X 取值为 1，Y 取值为 0，Z 取值为 1 时，代表这个状态的乘积项 $x\bar{y}z$，才为 1。为了简化，也可一个状态用一个二进制变量取值 1 代表它，0 不代表它。只是，这么处理须多用些变量。但这对内部期间丰富的 PLC 是可以接受的。

有向联线代表状态转换，要标注实现这个转换的当前脉冲信号输入（注在斜线的左方），还要标注这个转换后的输出（注在斜线的右方）。

图 3-41 即为一部分状态图。表示了有 3 个节点 Si－1、Si 及 Si＋1。Si－1 到 Si，由于输入 Xi 的作用。在此转换后，产生 Yi 的输出组合。Si 到 Si＋1，由于输入 Xi＋1 的作用。在此转换后，产生 Yi＋1 的输出组合。

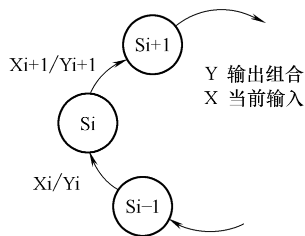


图 3-41　部分状态图

有了这些约定，可很容易建立起状态图与通电表间的对应关系。状态图的节点与通电表的节拍对应，通电表某节拍的当前输入，即为状态图的状态转换的脉冲信号输入，等等。

状态图可用以分析异步时序逻辑，也可用以设计异步时序逻辑。

3.4.2　同步时序逻辑分析

同步时序逻辑分析也与异步时序逻辑分析一样，也是研究设计好的程序。也是在“纸面”上“运行”程序。

具体方法是按照脉冲输入的实际顺序，逐一分析所有器件的状态，确定是 0、OFF、不工作，还是 1、ON、工作，进而弄清程序能否实现预定的功能。

显然，如果程序正确、可行，其输入、输出的对应关系也总是确定的。也就是说，对某个具体程序的分析，其答案只有一个。

以下为几个分析实例：

1. 显示刀架位置的同步时序逻辑（见图 3-4）

这里的 mW1、mW2、mW3 是对 WW1、WW2、WW3 的记录。实际两者的状态是相同的。所差的只是，不使 WW1、WW2、WW3 的变化在本脉冲周期起作用。因此，分析此逻辑时，可把 mW1、mW2、mW3 与 WW1、WW2、WW3 等同。

这样，状态可用 WW1、WW2、WW3 表示。分别为 100（WW1 ON）、010（WW2 ON）及 001（WW3 ON），共 3 个状态。状态间转换全靠脉冲信号 pA。根据置位、复位表达式，

可求得，在不同状态下，转换后的状态及其输出。图3-42所示的就是根据这些状态转换及输出画出的状态图。

图中还加入一个原始状态000（WW1、WW2、WW3均OFF）。它由初始脉冲pA0激发。目的是进入程序运行的第一个扫描周期，使WW1置位。

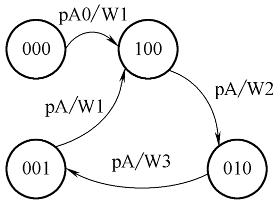


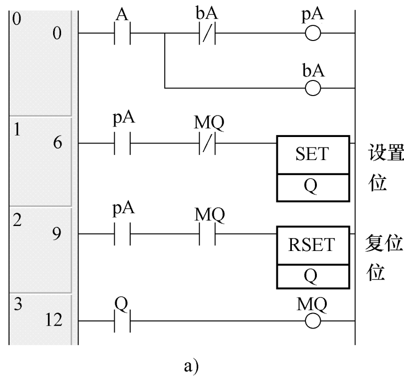
图3-42 显示刀架位置的同步时序逻辑状态图

2. 单按钮启、停电路

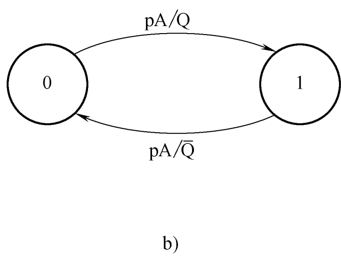
单按钮启、停电路，也可用S（置位）、R（复位）替代输出指令。替代后的同步化程序如图3-43所示。

这里，条0产生脉冲信号pA。条1、2为程序主体，当MQ OFF时，置位；而当MQ ON时，复位。而条3为MQ对Q状态的记录。实际两者的状态是相同的。所差的只是，不使Q的变化在本脉冲周期起作用。这也正是同步逻辑的特点。

可知，Q OFF时，经脉冲信号作用，将ON、置位；而在Q ON时，经脉冲信号作用，将OFF、复位。图3-43b所示的即是它的状态图。



a)



b)

图3-43 置位、复位指令单按钮启停
a）梯形图程序 b）状态图

3.“格兰码”计数器

图3-44所示为“格兰码”计数器。主令信号PP为脉冲信号（图中未列出它的产生）。B0、B1、B2为输出，而b00、b11、b22用以记录B0、B1、B2。

起始时，B2、B1、B0全为0，其状态用000表示。b22、b11、b00也全为0。这时，如计数脉冲PP ON，在执行计数段指令时，b22、b11、b00状态不会变，仍然全为0，从分析B0、B1、B2起动逻辑知，B0起动条件为1，将工作，ON。而B2、B1起动条件均为0，不可能工作，仍都为0。

之后，执行中间记录处的几条指令。执行后b00、b11、b22对B2、B1、B0的变化了的状态作了记录，为以后计数脉冲的作用提供新的逻辑条件。但不会对本扫描周期计数段指令的执行起作用。因为，到了下一个扫描周期，计数脉冲已OFF，B1、B2不可能起动。B0则由于计数脉冲的非为1，保持条件为1，肯定将继续ON。可知，000状态，接受一个计数脉冲后，将变为001。

在 001 时，若计数脉冲 PP 再 ON，在执行计数段指令时，b22、b11、b00 状态不变，为 001，从分析知，B0 保持条件为 1，将保持工作，仍为 ON。而 B1 起动条件为 1，将起动、工作，变为 ON。而 B2 起动条件为 0，不能工作，仍为 0。之后，执行中间记录处的几条指令。执行后，b22、b11、b00 将记录为 011。又为下一次计数作准备。可知，001 状态，接受一个计数脉冲 PP 后，将变为 011。

在 011 时，若计数脉冲 PP 再 ON，在执行计数段指令时，b22、b11、b00 状态不变，为 011，从分析知，B0 保持条件为 0，起动条件也为 0，B0 将停止工作，转为 OFF。而 B1 保持条件为 1，将保持工作，仍为 ON。而 B2 起动条件为 0，不能工作，仍为 0。之后，执行中间记录处的几条指令。执行后，b22、b11、b00 将记录为 010。又为下一次计数作准备。可知，011 状态，接受一个计数脉冲后，将变为 010。

在 010 时，若计数脉冲 PP 再 ON，还可作类似的分析，可知它将为 110。再接着为 111。再接着为 101。再接着为 100。再接着为 000。

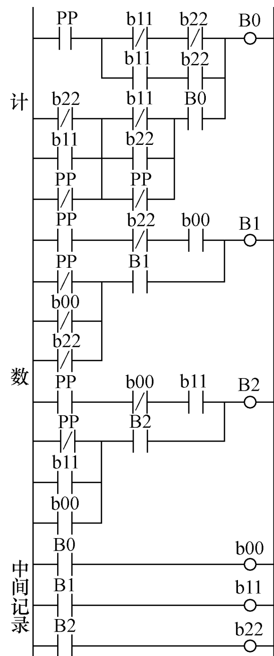


图 3-44　“格兰码”计数器

把上述分析，按状态图的约定，即可得出它的状态图，如图 3-45 所示。

从状态图分析可知，这里状态的变化正是按“格兰码”计数器的计数规律变化的。

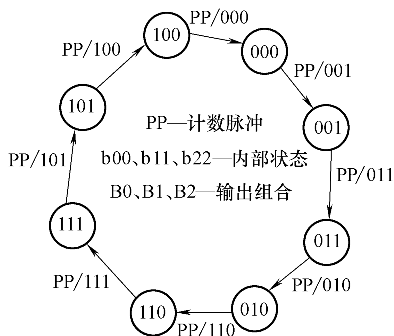


图 3-45　“格兰码”计数器状态图

3.4.3　同步时序逻辑综合

梯形图逻辑状态图法综合的目的也是，要编写出满足设计要求，并受设计条件约束的 PLC 程序，是分析的反问题。

用状态图法，综合的步骤是：

（1）信号分析：确定有效主令信号及反馈信号，并使其变为所需要的脉冲信号。

（2）状态分析：确定基本工作状态、可能工作状态及不使用状态，及这些状态间的转换与转换条件。

（3）画状态图：按确定的状态及其转换，初画状态图。

（4）简化状态图：依据状态图化简原则，简化状态图。简化的原则是，两个或多个状态，如原输出相同，进入次一状态的信号及产生的输出又相同，则可合并、视为同一状态。

（5）定变量：确定用以表达状态的变量及其编码，确定输出变量。

（6）列表达式：依各变量在状态图中的关系，列写各变量的逻辑表达式，并化简各表达式。

（7）画梯形图：依据化简后的表达式画出梯形图。

以下用实例说明状态图法综合的过程。

3.4.4 同步时序逻辑状态图法设计实例

1. 旋转自动门工作控制程序

要求是：当电源开关处于 ON 时，进入工作状态；反之，退出工作状态。在工作状态下，如检测到门前有人，则自动门正转；如检测不到门前有人，经若干延时，并到达给定位置后自动门停转。在正转时，如检测到闯人或夹人，则反转；如检测到不再闯人或夹人，经若干延时，自动门又恢复正转。

（1）信号分析

主令信号有：

电源开关信号，对应的有电源开关 ON 脉冲信号（Pd）及电源开关 OFF 脉冲信号（Pdn）；

反馈信号有：

检测门前是否有人，用光电开关。对应的有，检测到门前有人光电开关 ON 脉冲信号（Py），检测到门前有人光电开关 OFF（门前没人）延时脉冲信号（Pyn）；

检测到门是否闯人或夹人，用光电开关。对应的有，检测到闯人或夹人光电开关 ON 脉冲信号（Pc），检测到闯人或夹人光电开关 OFF（不闯人或夹人）延时脉冲信号（Pcn）；

自动门到位与否，用行程开关。对应的有，自动门到位脉冲信号（Pw）。

（2）状态分析

1）基本工作状态有：

状态 0：自动门不工作。如检测到电源开关 ON 脉冲信号（Pd），则进入状态 1。

状态 1：自动门处工作状态。但既不正转，也不反转。如检测到门前有人光电开关 ON 脉冲信号（Py），则进入状态 2。

状态 2：自动门正转。如检测到闯人或夹人光电开关 ON 脉冲信号（Pc），则进入状态 4。如检测到门前有人光电开关 OFF（门前没人）延时脉冲信号（Pyn），则进入状态 3。

状态 3：自动门正转。如检测到自动门到位脉冲信号（Pw），则进入状态 1；如又检测到门前有人光电开关 ON 脉冲信号（Py），则又回到状态 2；

状态 4：自动门反转。如检测到闯人或夹人光电开关 OFF（不闯人或夹人）延时脉冲信号（Pcn），则又回到状态 2。

2）可能状态有：

状态 4A、状态 4AA：处状态 4 时，自动门反转，但这时可能检测到 Pyn 信号。对此要用一个状态（状态 4A）作记录，即从 4 转到 4A，自动门仍应反转。这时，如检测到 Pcn 信号，也要用一个状态（状态 4AA）对此作记录，但自动门仍正转。在状态 4AA 时，如接收到 Pw 信号，则转为状态 1，自动门工作，但停转；如检测到 Pc 信号，那又要回到状态 4A，自动门又反转。

状态 1A、状态 1AA：处状态 1 时，自动门处工作状态。但这时可能检测到 Pc 信号。这可用一个状态（状态 1A）对此应作记录，并使自动门反转。这时，如检测到 Pcn 信号，则也要用一个状态（状态 1AA）作记录，自动门转为正转。在状态 1AA 时，如接收到 Pw 信号，则转为状态 1，自动门工作，但停转；如检测到 Pc 信号，那又要回到状态 1A，自动门又反转。

状态 1AAA：处状态 1A 时，也可能检测到 Py 信号。故要用一个状态（状态 1AAA）作记录，自动门仍应反转。直到检测到 Pcn 信号时，转到状态 2，自动门正转。当然，在 1AAA 时，也可能又检测到 Pyn 信号，那又要回到状态 1A。

另外，在所有状态下，只要检测到电源开关 OFF 脉冲信号（Pdn），则进入状态 0，自动门不工作。

总之，自动门所有可能的状态都要考虑到，不能遗漏。应避免有的有效信号得不到反应或记录，那样，PLC 将不可能所要求的实现控制。

（3）画状态图

根据以上状态的描述，初画出如图 3-46 所示的状态图。为了清晰，该图未把在任何状态下，如检测到 Pdn 信号，都转到状态 0 画出（只把状态 1，检测到信号 Pdn 转到状态 0 画出）。

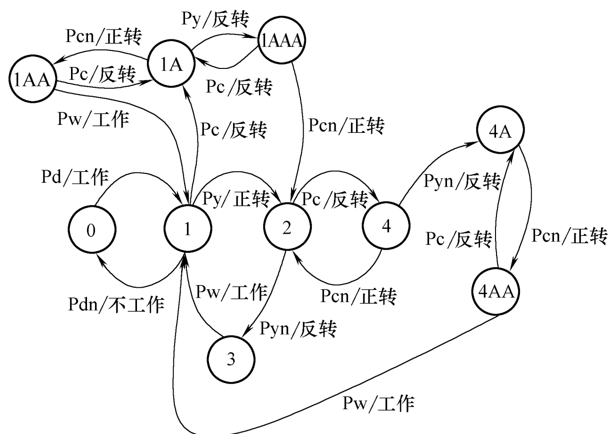


图 3-46　初画自动门控制状态图

对初画的状态图可进行化简。其原则是：相同信号进入相同次状态及产生相同输出的两个或多个状态，可合并，可视为同一状态。如图 3-46 所示的状态 1AA、4AA 与状态 3，可视为同一状态，可合并。状态 4 及状态 1AAA 也可视为同一状态，也可合并。经合并后的状态图如图 3-47 所示。

再观测图 3-47 看出，状态 1A 与状态 4A 仍可合并，合并后如图 3-48 所示。到此没有可合并的状态了，因而，也就完成了状态图的设计。

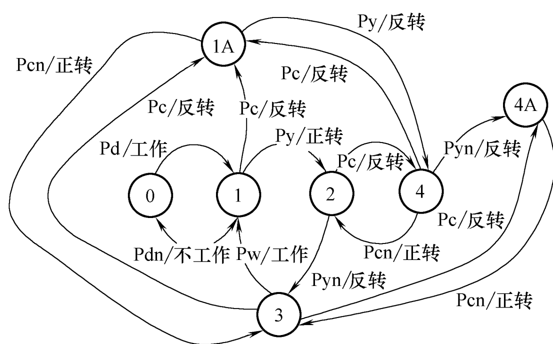


图 3-47　自动门控制状态图化简

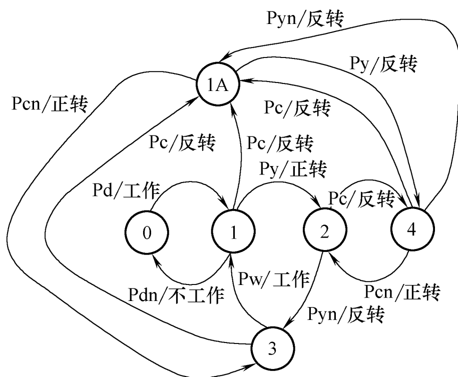


图 3-48　化简后自动门控制状态图

从本例可知，状态图设计与通电表惟一性设计是不同的。状态图设计是状态化简，逐步减少状态；而通电表设计是避免逻辑条件相混，逐步增加内部变量（在某种意义上讲，也可说是增加状态）。同时，状态图的化简不是必要的，也可不化简。这时，只是内部变量要多用一些，但仍可实现所要求的控制。而通电表惟一性设计则是必要的，惟一性原则不满足，将不能实现所要求的逻辑控制。

（4）确定变量

1）状态变量：本例共有 6 个状态，用 3 个内部变量 M3、M2、M1 的不同取值，即可把这 6 个状态分开。如本例各状态用二进制编码：

000（M3 = 0、M2 = 0、M1 = 0）代表状态 0；

001（M3 = 0、M2 = 0、M1 = 1）代表状态 1；

010（M3 = 0、M2 = 1、M1 = 0）代表状态 2；

011（M3 = 0、M2 = 1、M1 = 0）代表状态 3；

100（M3 = 1、M2 = 0、M1 = 0）代表状态 4；

101（M3 = 1、M2 = 0、M1 = 1）代表状态 1A；

2）输出变量：本例的输出变量有：

工作，用输出点 10.00 控制；

正转，用输出点 10.01 控制；

反转，用输出点 10.02 控制。

没有以上 3 个输出，即为不工作。

（5）列写逻辑表达式

有了化简后状态图及状态与输出变量，就可列写逻辑表达式。

1）状态变量逻辑表达式

M1 置位：在状态 1、3、及 1A 为 1，从其它状态进入此状态应置位。从图 3-48 知须置位的有：状态 0 到状态 1（Pd 作用）；状态 2 到状态 3（Pyn 作用）；状态 4 到状态 1A（Pc 或 Pyn 作用）。

具体表达式为

SET(M1) = (000) * Pd + (010) * Pyn + (100) * (Pc + Pyn)

M1 复位：状态状态 0、2 及 4 为 0，从其它状态进入此状态应复位。从图 3-48 知须复位的有：状态 1 到状态 2（Py 作用）；状态 1A 到状态 4（Py 作用）。再有是检测到 Pdn 时，也要复位。表达式为

RSET(M1) = (001) * Py + (101) * Py + Pdn

M2 置位：在状态 2、3 为 1，从其它状态进入此状态应置位。从图 3-48 知须置位的有：状态 1 到状态 2（Py 作用）；状态 4 到状态 2（Pcn 作用）；状态 1A 到状态 3（Pcn 作用）。

具体表达式为

SET(M2) = (001) * Py + (100) * Pcn + (101) * Pcn

M2 复位：状态 0、1、4 及 1A 为 0，从其它状态进入此状态应复位。从图 3-48 知须复位的有：状态 2 到状态 4（Pc 作用）；状态 3 到状态 1（Pw 作用）；状态 3 到状态 1A（Pc 作用）。再有是检测到 Pdn 时，也要复位。表达式为

RSET(M2) = (010) * Pc + (011) * Pw + (011) * Pc + Pdn

M3 置位：在状态 4 及 1A 为 1，从其它状态进入此状态应置位。从图 3-48 知：须置位的有：状态 3 到状态 1A（Pc 作用）；状态 2 到状态 4（Pc 作用）；状态 1 到状态 1A（Pcn 作用）。

具体表达式为

$$SET(M3)=(011)*Pc+(010)*Pc+(001)*Pc$$

M3 复位：状态状态 0、1、2 及 3 为 0，从其它状态进入此状态应复位。从图 3-48 知须复位的有：状态 1A 到状态 3（Pcn 作用）；状态 4 到状态 2（Pcn 作用）。再有是检测到 Pdn 时，也要复位。表达式为

$$RSET(M3)=(101)*Pcn+(100)*Pcn+(001)*Pc+Pdn$$

2）输出变量逻辑表达式

正转 $=10.00=\overline{M3}*M2*M1+\overline{M3}*M2*\overline{M1}$

反转 $=10.01=M3*\overline{M2}*M1+M3*\overline{M2}*\overline{M1}$

工作 $=10.02=\overline{M3}*\overline{M2}*M1$

这式子可进行化简，具体化简过程在此不做介绍。以下的梯形图，如图 3-49 所示，是依未化简的的逻辑关系画出的。经调试说明此设计是可行的。只是，这里未把脉冲信号的梯形图画出。

（6）画梯形图

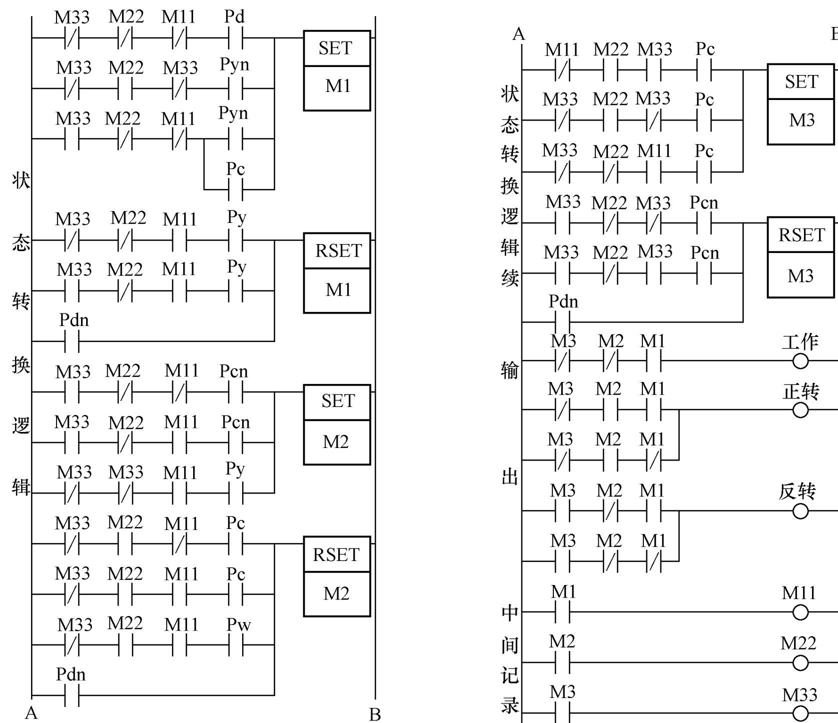


图 3-49　自动门控制梯形图程序

图 3-49 是依据上述逻辑表达式画出的，有 3 个部分，一为状态转换逻辑；二为输出逻辑；三为中间记录逻辑。图中 A 与 A、B 与 B 是相接的。显然，如经逻辑化简，该梯形图可得到很大的简化，这里略。

2. 图 3-1 所示的液压扳动手柄程序设计

要求是：不管手柄处在何位置，按下 A1 后，应使其转到使 XK1 合下的位置；按 A2，则到使 XK2 合下的位置；按 A3，则到使 XK3 合下的位置。

（1）信号分析

主令信号有：

A1、A2、A3 ON 脉冲信号（pA1、pA2、pA3）；

反馈信号有：

XK1、XK2、XK3 ON 脉冲信号（pXK1、pXK2、pXK3）。

（2）状态分析

1）基本工作状态有：

状态 0：手柄处于 XK1 压下位置，手柄不转。检测到 pA2，进入状态 1，手柄顺时针转。检测到 pA3，进入状态 1A，手柄顺时针转。

状态 1：手柄顺时针转，到 XK2 压下位置，检测到 pXK2，进入状态 2，手柄停转。

状态 1A：手柄顺时针转，到 XK3 压下位置，检测到 pXK3，进入状态 4，手柄停转。

状态 2：手柄处于 XK2 压下位置，手柄不转。检测到 pA3，进入状态 3，手柄顺时针转。检测到 pA1，进入状态 3A，手柄逆时针转。

状态 3：手柄顺时针转，到 XK3 压下位置，检测到 pXK3，进入状态 4，手柄停转。

状态 3A：手柄逆时针转，到 XK1 压下位置，检测到 pXK1，进入状态 0，手柄停转。

状态 4：手柄处于 XK3 压下位置，手柄不转。检测到 pA2，进入状态 5，手柄逆时针转。检测到 pA1，进入状态 5 A，手柄逆时针转。

状态 5：手柄逆时针转，到 XK2 压下位置，检测到 pXK2，进入状态 2，手柄停转。

状态 5A：手柄逆时针转，到 XK1 压下位置，检测到 pXK1，进入状态 0，手柄停转。

2）其它可能的状态没有了。

（3）画状态图

根据以上状态的描述，初画出如图 3-50 所示的状态图。为了清晰，该图未把在任何状态下，如检测到 Pdn 信号，都转到状态 0 画出（只把状态 1，检测到信号 Pdn 转到状态 0 画出）。

对初画的状态图可进行化简。如图 3-50 的状态 1A 与状态 3，可视为同一状态，可合并。状态 3A 状态 5A 也可视为同一状态，也可合并。经合并后的状态图如图 3-51 所示。

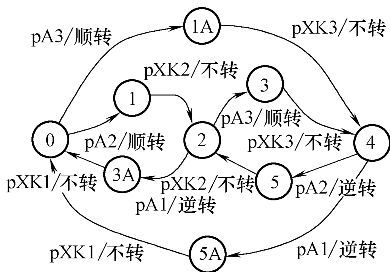


图 3-50 液压扳动手柄控制状态图

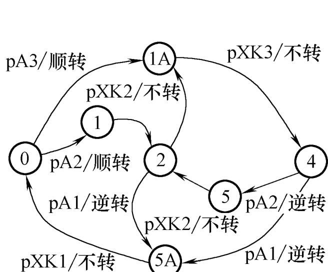


图 3-51 液压扳动手柄控制状态图化简

从本例也可知，状态图设计与通电表惟一性设计是不同的。状态图设计是状态化简，逐步减少状态；而通电表设计是避免逻辑条件相混，逐步增加内部变量（在某种意义上讲，也可说是增加状态）。同时，状态图的化简不是必要的，也可不化简。这时，只是内部变量

要多用一些，但仍可实现所要求的控制。而通电表惟一性设计则是必要的，惟一性原则不满足，将不能实现所要求的逻辑控制。

（4）确定变量

1）状态变量：本例共有 7 个状态，为了简化，每个状态用一个内部变量 MM1、MM2、MM3、MM4、MM5、MM6、MM7 取值 1，分别代表状态 0、1、1A、2、4、5、5A。

2）输出变量：本例的输出变量有

顺时针转，用输出点 10.00 控制。逆时针转，用输出点 10.01 控制。没有以上 3 个输出，即为不转。

（5）列写逻辑表达式

有了化简后状态图及状态与输出变量，就可列写逻辑表达式。

1）状态变量逻辑表达式

SET(MM1) = (MM7) *pXK1 + XK1 *P_First...（此项考虑初始化加入）

RSET(MM1) = (MM1) *(pA2 + *pA3)

SET(MM2) = (MM1) *pA2

RSET(MM2) = (MM2) *pXK2

SET(MM3) = (MM1) *(pA2 + pA3)

RSET(MM3) = (MM2) *(pXK2 + pXK3)

SET(MM4) = (MM2 + MM6) *pXK2 + XK1 *P_First...（此项考虑初始化加入）

RSET(MM4) = (MM4) *(pA1 + pA3)

SET(MM5) = (MM3) *pXK3 + XK3 *P_First...（此项考虑初始化加入）

RSET(MM5) = (MM3) *(pA2 + pA1)

SET(MM6) = (MM5) *pA2

RSET(MM6) = (MM6) *pXK2

SET(MM7) = (MM4 + MM5) *pA1

RSET(MM7) = (MM7) *pXK1

2）输出变量逻辑表达式

顺时针转，10.00 = MM2 + MM3

逆时针转，10.01 = MM6 + MM7

（6）画梯形图

图 3-52 是依据上述逻辑表达式画出的，有 3 个部分，条 6 到条 19 为状态转换逻辑；条 20、21 为输出逻辑；条 22 到条 28 为中间记录逻辑。微分信号生成略。显然，如经逻辑化简，该梯形图可简化，这里略。

3. 组合机床动力头运动控制

本例的工艺要求同本章第 3 节例 3。它的输出有两个变量，进与慢，用 DT1、DT2 表示。DT1 ON 进、DT2 ON 慢，反之为退、快。

依题意，主轴在原位时（1XK 合上），给出起动信号（用脉冲信号 Q），主轴快（DT1、DT2 取值为 10），以使主轴快速接近工件。

主轴快进到 2XK 合上（这时 1XK 早已松开），则利用它的脉冲作用使主轴转为慢进（DT1、DT2 取值为 11），以进行加工。

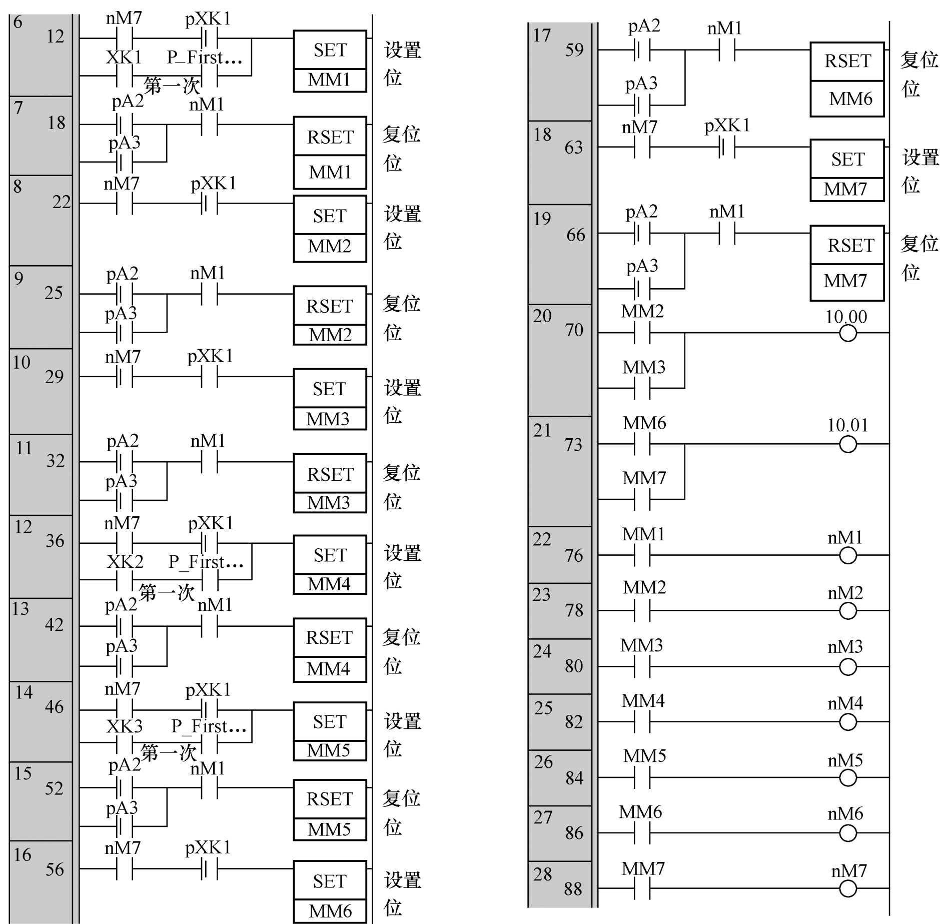


图 3-52 手柄转动控制梯形图程序

主轴加工前进到 3XK 合上，（这时 2XK 也早已松开），则利用它的脉冲作用使主轴转为快退（DT1、DT2 取值为 00），以便于钻头排屑。

主轴后退到 2XK 合上（这时 3XK 早已松开），再利用它的脉冲作用使主轴转为快进（DT1、DT2 取值为 10），使主轴接近工件未加工的部分。

主轴快进到 3XK 合上（这时 2XK 早已松开），则利用它的脉冲作用使主轴转为慢进（DT1、DT2 取值为 11），以进行未加工的部分的加工。

主轴加工前进到 4XK 合上，（这时 3XK 也早已松开），则利用它的脉冲作用使主轴转为快退（DT1、DT2 取值为 00），直到主轴回到原位，1XK 合上，完成一个工作循环。

按以上要求，用状态图法设计如下：

（1）信号分析

1）有效主令信号为起动信号 pQ。

2）反馈信号有动力头到位行程开关 1XK、2XK、3XK、4XK ON 发出的，相应脉冲信号 p1XK、p2XK、p3XK、p4XK。及 1XK、2XK、3XK、4XK OFF 发出的，相应脉冲信号

p1OFF、p2OFF、p3OFF、p4OFF。

（2）状态分析

状态 0：动力头处原位，1XK ON，其它开关 2XK、3XK、4XK OFF。输出 DT1、DT2 为 00。在此状态时，按下起动按钮 Q，接收相应的脉冲信号 pQ 后，进入状态 1。

状态 1：动力头第一次快进。输出 DT1、DT2 为 11。这时，接收到脉冲信号 p2XK，则进入状态 2。

状态 2：动力头第一次工进。输出 DT1、DT2 为 10。这时，接收到脉冲信号 p3XK，则进入状态 3。

状态 3：动力头第一次快退。输出 DT1、DT2 为 01。这时，接收到 2XK OFF 脉冲信号 p2OFF，则进入状态 4。

状态 4：动力头第二次快进。输出 DT1、DT2 为 11。这时，接收到脉冲信号 p2OFF，则进入状态 5。

状态 5：动力头第二次工进。输出 DT1、DT2 为 10。这时，接收的 p4XK 脉冲信号，则进入状态 6。

状态 6：动力头第二次快退。输出 DT1、DT2 为 01。这时，接收的 p1XK 脉冲信号，则进入状态 0。

（3）画状态图。按确定的状态及其转换，画出的状态图如图 3-53 所示。

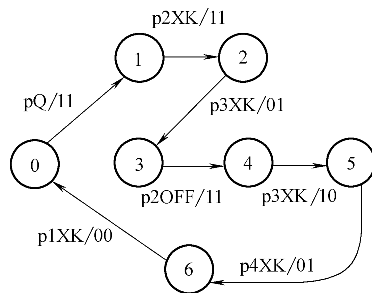


图 3-53　动力头运动控制状态图

（4）确定变量

图 3-53 有 7 个状态，为了简化，这里用分别用 M1、M2、M3、M4、M5、M6 等 7 个变量代表它。哪一个 ON，即进入相应的那一个状态。这些变量全为 OFF 即为状态 0。也可用动力头在原位，即 1XK ON 代表状态 0。

（5）列表达式

M1 置位：在状态 1 时为 1。从图 3-53 知，在状态 0 时，经 pQ 作用，应置位，进入本状态。

具体表达式为

$$SET(M1) = 1XK * pQ$$

M1 复位：本状态为 1，其它状态均为 0。从图 3-53 知，经 p2XK 作用，进入状态 2，应复位。

具体表达式为

$$RSET(M1) = M1 * p2XK$$

M2 置位：在状态 2 时为 1。从图 3-53 知，在状态 1 时，经 p2XK 作用，应置位，进入本状态。

具体表达式为

$$SET(M2) = M1 * p2XK$$

M2 复位：本状态为 1，其它状态均为 0。从图 3-53 知，经 p3XK 作用，进入状态 3，应复位。

具体表达式为

$$\mathrm{RSET}(M2)=(M2)*p3XK$$

M3 置位：在状态 3 时为 1。从图 3-53 知，在状态 2 时，经 p3XK 作用，应置位，进入本状态。

具体表达式为

$$\mathrm{SET}(M3)=M2*p3XK$$

M3 复位：本状态为 1，其它状态均为 0。从图 3-53 知，经 p2OFF 作用，进入状态 4，应复位。

具体表达式为

$$\mathrm{RSET}(M3)=(M3)*p2OFF$$

M4 置位：在状态 4 时为 1。从图 3-53 知，在状态 3 时，经 p2OFF 作用，应置位，进入本状态。

具体表达式为

$$\mathrm{SET}(M4)=M3*p2OFF$$

M4 复位：本状态为 1，其它状态均为 0。从图 3-53 知，经 p3XK 作用，进入状态 5，应复位。

具体表达式为

$$\mathrm{RSET}(M4)=(M4)*p3XK$$

M5 置位：在本状态时为 1。从图 3-53 知，在状态 4 时，经 p4XK 作用，应置位，进入本状态。

具体表达式为

$$\mathrm{SET}(M5)=M4*p3XK$$

M5 复位：本状态为 1，其它状态均为 0。从图 3-53 知，经 p4XK 作用，进入状态 6，应复位。

具体表达式为

$$\mathrm{RSET}(M5)=(M5)*p4XK$$

M6 置位：在本状态时为 1。从图 3-53 知，在状态 5 时，经 p4XK 作用，应置位，进入本状态。

具体表达式为

$$\mathrm{SET}(M6)=M5*p4XK$$

M6 复位：本状态为 1，其它状态均为 0。从图 3-53 知，经 p1XK 作用，进入状态 0，应复位。

具体表达式为

$$\mathrm{RSET}(M6)=(M6)*p3XK$$

（6）画梯形图。依化简后的逻辑表达式画出梯形图。

根据所列表达式，画出梯形图如图 3-54 所示。该图未画出脉冲生成逻辑。仅画了状态转换逻辑及输出逻辑。

在状态转换逻辑中，M1～M6 是逆序安排的。目的是实现，在脉冲作用期间“前后逻辑条件一致”（见本章 3.4.3 节）。否则状态转换逻辑无法实现。

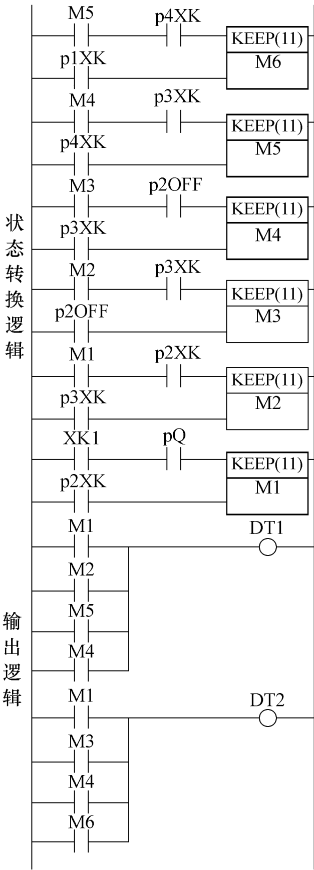


图 3-54　动力头运动控制梯形图程序

3.5　图解法编程

关键词：时序图、时序图法、时间区间、流程图、流程图法、顺序流程序、条件分支程序、平行分支流程、循环程序

3.5.1　时序图法编程

1. 时序图法设计要点

时序图（Timing Diagrams），是信号随时间变化的图形。横坐标为时间轴，纵坐标为信号值，其取值为 0 或 1。以这种图形为基础，进行 PLC 程序设计，称时序图法。

时序图是从用示波器分析一些电器硬件工作过程，引申出来的，用它可分析与确定有关逻辑量间的时序关系。

这个方法的设计步骤为

（1）画时序图。根据要求，画输入、输出信号的时序图，建立起准确的时间对应关系。

（2）确定时间区间。找出时间的变化临界点，即输出信号应出现变化的点，并以这些点为界限，把时段划分为若干时间区间。

（3）设计定时逻辑。可用多个定时器，建立各个时间区间；也可用秒脉冲计数器，记录时间，然后，再通过比较，建立各个时间区间。

（4）确定动作关系。根据各动作与时间区间的对应关系，建立相应的动作逻辑，列出各输出变量的逻辑表达式。这可按输出要求进行的设计，一般为组合逻辑的问题，是不难实现的。

（5）画梯形图。依定时逻辑及输出逻辑的表达式，画梯形图。

用时序图法设计的前提为，输入与输出间有对应的时间顺序关系，其各自的变化是按时间顺序展开的。显然，不满足这个前提，无法画时序图，也无从用这个方法设计了。

以下举两个例子，说明如何用本法进行设计：

2. 时序图法设计实例

（1）设计喷泉电路。要求设计一个控制喷泉工作的电路。喷泉有 A、B、C 三组喷头，如图 3-55a 所示。工作过程应按图 3-55b 所示，即：起动后，A 组先喷 5s，后 B、C 同时喷，5s 后 B 停，再 5s C 停，而 A、B 又喷，再 2s，C 也喷。持续 5s 后全部停喷。再 3s A 又重复前述过程。

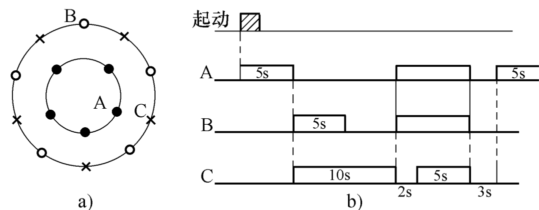


图 3-55　喷泉控制简图

首先，分配入、出触点。用 0002 接起动按钮，0003 接停止按钮；而 A、B、C 分别用 10.00、10.01、10.02 控制。

其次，是设计程序。它是定时控制程序，较适合用本法设计。

1）画出时序图。按设计要求，画输出时序图，如图 3-55b 所示。

2）确定定时区间。从图知，它有 7 个临界点，要有 6 个时间区间，用以作有关输出控制。

3）设计定时逻辑。本例用 6 个定时器，建立各个时间区间。

这 6 个定时器（TIM000 ~ TIM005）工作时序如图 3-56 所示。

其工作过程是：0.00 ON，起动 10.00。进而使定时器 TIM0 工作。延长 5s 后，又使定时器 TIM1 工作……直到 TIM5 工作后，延时 3s，使 TIM0 OFF，进而使 TIM1 OFF，再进而使 TIM2 OFF……，直到使 TIM5 自身 OFF，定时逻辑复原。复原后，如无停止信号，将又是 TIM0 工作，又开始新的循环。

与此对应的梯形图如图 3-57a 所示。

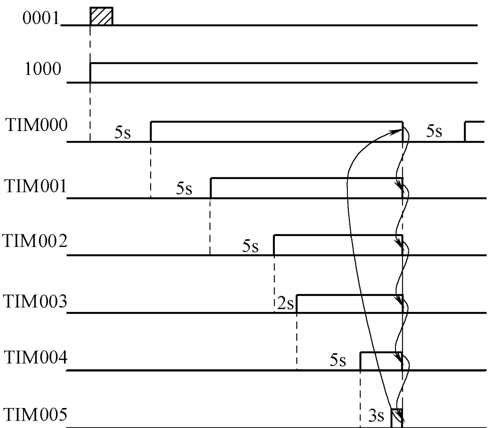


图 3-56　喷泉控制时序图

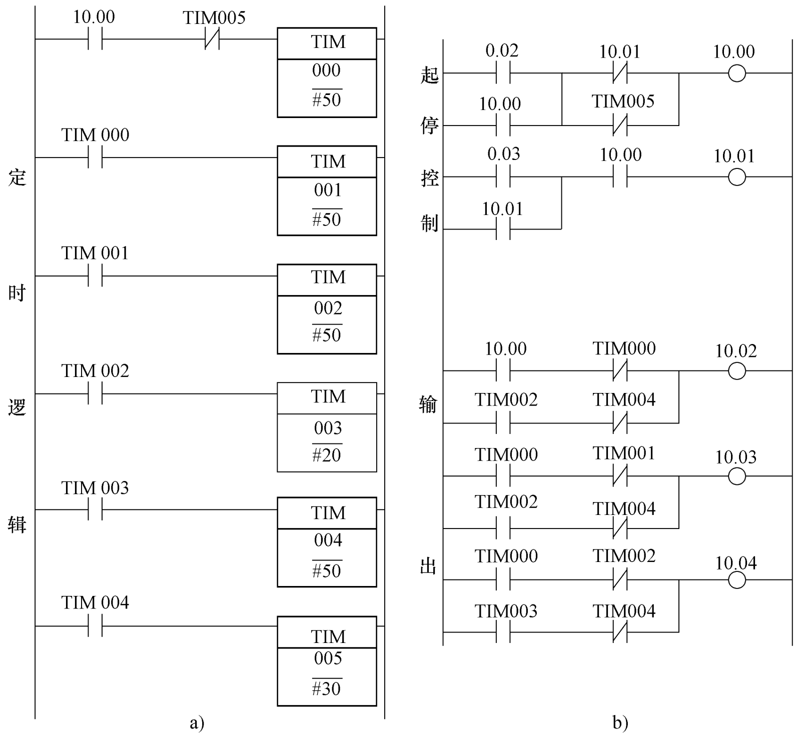


图 3-57　喷泉控制梯形图程序

4）确定动作关系。从图 3-57a 知，它有 6 个时间区间。这些区间及其逻辑条件为

区间 1，其逻辑条件为：10.00 · $\overline{\text{TIM000}}$

区间 2，其逻辑条件为：TIM000 · $\overline{\text{TIM001}}$

区间 3，其逻辑条件为：TIM001 · $\overline{\text{TIM002}}$

区间 4，其逻辑条件为：TIM002 · $\overline{\text{TIM003}}$

区间 5，其逻辑条件为：TIM003 · $\overline{\text{TIM004}}$

区间 6，其逻辑条件为：TIM004

在这 6 个区间中，A、B 及 C 的取值，如图 3-55 所示。依图可列出输出变量 A（10.02）、B（10.03）及 C（10.04）的逻辑表达式。具体是：

$$10.02 = 10.00 * \overline{TIM000} + TIM002 * \overline{TIM004}$$

$$10.03 = TIM000 * \overline{TIM001} + TIM002 * \overline{TIM004}$$

$$10.04 = TIM000 * \overline{TIM002} + TIM003 * \overline{TIM004}$$

5）画梯形图。与这个定时逻辑及输出逻辑对应的梯形图，如图 3-57b 所示。这里，0002 为起动信号，它 ON 后可使 10.00 ON，并自保持。此即开始了周而复始的定时控制。

开始工作后，若按下结束工作循环按钮，这将使 0.03 ON。0.03 ON 后，由于 1001 有自保持，可保持 ON 状态。到了 TIM005 ON，其常闭触点 $\overline{\text{TIM005}}$ 使 10.00 OFF。10.00 OFF 后，TIM000 将不再工作，整个循环停止。同时，10.00 的 OFF 也将使 10.01 OFF，整个电路复原。

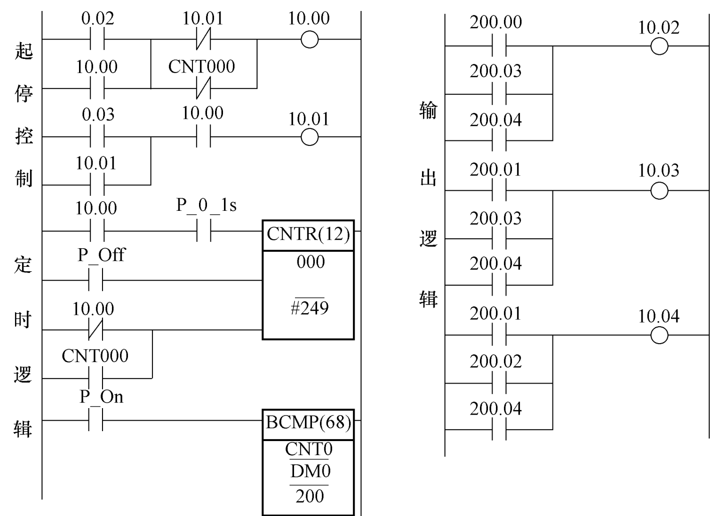


图 3-58　又一喷泉控制梯形图程序

图 3-58 也是一个实现上述逻辑的梯形图。该图的定时逻辑由计数器 CNTR 000 加 0.1s 的定时脉冲建立。从图知，0.02 ON 后，使 10.00 ON，并自保持。CNT 000 计数，每 0.1s，计数器的现值加 1。加到 249。再加 1 时，其现值恢复为 0000，且常闭触点 OFF。

这时，如果 0.03 未按下（仍为 OFF），10.00 仍保持，CNT 000 又从 0000 开始计数。循环又重复进行。如果 0.03 按下（要停循环），则 CNT 000 的常闭触点，将使 10.00 OFF。CNT 000 将不再计数，循环即可停止。

这里 CNT 000 的计数值可用来进行时间区间的划分。CNT000 值 0 ~ 49，即对应时间段 1。CNT000 值 50 ~ 99，即对应时间段 2。等等。

图 3-58 设了一个范围比较指令，把这个时间对应输出于 200 通道的对应位。比较范围值放置在 DM0 ~ DM31 中。依本例，DM0 ~ DM31 的值分别为

DM00	0000	DM01	0049
DM02	0050	DM03	0099
DM04	0100	DM05	0149

DM06	0150	DM07	0169
DM08	0170	DM09	0219
DM10	0220	DM11	0249

其它的，即 DM12 ~ DM31 也要被范围比较指令使用，但不起作用。

与这 6 个范围对应输出位，依次为 200.00 ~ 200.05。即若计数值落在 0000 ~ 0049 范围内，200.00 ON，其它的均 OFF。若落在 0050 ~ 0099 范围内，则 200.01 ON，其它的均 OFF……，显然，这里的 200.00，即上述的 $10.00\cdot\overline{\text{TIM000}}$；200.01，即上述的 $\text{TIM000}\cdot\overline{\text{TIM001}}$……

依这个对应关系，画出的输出逻辑如图 3-58 所示。

（2）设计一个有顺序要求的一组设备起动停车程序。要求起动时，依次使若干设备起动起来，如每个设备起动间隔 10s。而停车也是依次进行，但顺序倒过来，先起动的最后停车。电厂的输煤系统的若干传送带的工作，常是这么要求的。

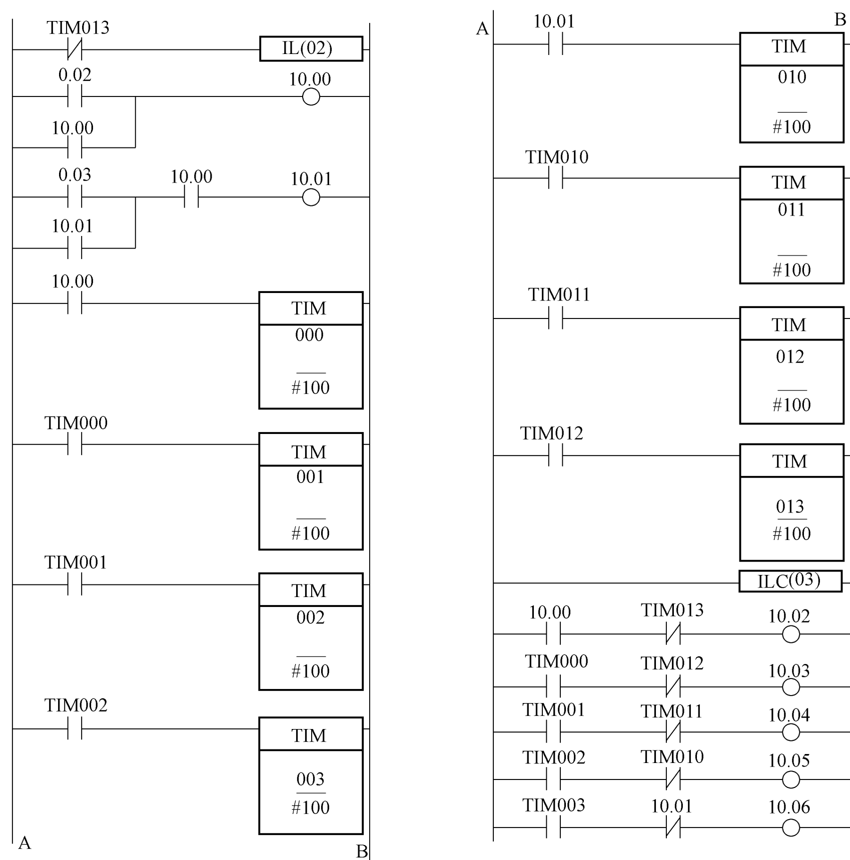


图 3-59 顺序起、逆序停梯形图程序

图 3-59 为设计好的程序。它的起动按钮号为 0.02，停止按钮号为 0.03。以这两个信号为输入，起动两组时间继电器（TIM000 ~ TIM003 及 TIM010 ~ TIM013）工作。分别产生起动及停车时间序列。

整个时间逻辑以互锁指令保持，当停止时的最后一个时间继电器工作（TIM013 ON），

则使 IL 的条件 OFF，这将使电路复原。

而在停止时间序列的起动条件中，加入已完成起动的条件（10.00 ON），目的是只有起动后，按停车按钮才有效。

有了这两组时间序列，其起动与时间的对应关系，是组合逻辑问题，可依条件直接列出。本例以此画出的梯形图如图 3-59 所示。

应当指出，时序图法适合于定时，计数控制，但不等于只能用于这种控制。没有定时，只是时间区间不是定值，而是随机值。若用计数器的值区分时，不必计时间脉冲，经一临界点，也可使计数值增 1。然后用表比较（不是范围比较）再把输出状态确定，也可很方便地进行设计。

3.5.2　流程图法编程

1. 流程图法设计要点

流程图，用以表示系统工作过程的图形。由一系列框图、圆圈及其连线组成，如图3-60所示。方框代表动作；圆圈代表动作起始位与动作终了；连线代表流向；短横线代表一个动作到另一个动作转换的条件。此外，有时也有分支流程，不同条件有不同的流向。这种图，可以把控制对象的工作过程清楚地表示出来，是常用的计算机程序设计的方法。

流程图法要用到步指令，PLC 多都有步指令。如实在无步指令，也可用移位、或基本的逻辑指令实现。

（1）常见流程图

1）顺序流程：图 3-60a 为顺序流程序。其进程是确定的，只有前一步骤进行后，建立了相应条件，其后续才可继而进行。

2）条件分支流程：图 3-60b 为条件分支程序。它依条件不同而改变程序流程。这对应于随机电路。

3）平行分支流程：图 3-60c 为平行分支程序。它依条件可进入平行的两个程序流程。

此外，还可能有循环流程。它的进程达到终点时，又返回到起点。

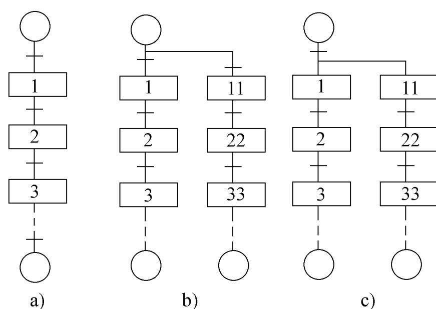


图 3-60　三种流程图

流程图法适合于动作步骤清晰的系统的控制逻辑设计。它的每一步对应一个动作，而步的转换，则看条件是否满足。所以，流程图画出来了，要求有多少步也就清楚了，而步的转换，有了条件也好设计。

（2）流程图法设计步骤

1）划分动作步。依控制对象的工作过程，划分动作步，画动作顺序图。

2）分配输入输出地址。分配与动作对应的输出地址。分配与步转换条件对应的输入地址。

3）设计流程图。这是动作顺序图的深入。要把 I/O 点号与动作顺序图的步建立联系。方框，即动作与 O（输出）联系，而条件，即短横线与 I（输入）联系。这个图应尽可能详尽，以便于进一步设计。

4）建立步进逻辑程序。如果 PLC 有步进指令，可用它建立这个程序。若没有，可用移

位指令，或直接用逻辑控制指令建立。

5）建立步与动作的关系。根据动作要求，一个步一个步地列写输出表达式，并依表达式画梯形图。

用流程图法设计的前提为所处理对象的工作步骤要比较清晰，便于划分；步转换的条件也较明确，便于建立逻辑关系。

2. 流程图法设计实例

（1）机械手控制程序设计。图 3-61 所示为机械手工作简图。

其工作过程应是：机械手向下，直到 A 点——夹取工件——上升——右进——下降，直到在 B 点——松开工件——上升——左退到原位，停止工作。若夹取不到工件，则从 A 点上升后，不右进，并报警，提示无工件。其入出点的编号参见设计过程，这里略。

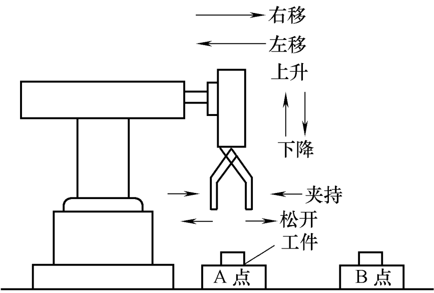


图 3-61 机械手工作简图

1）划分动作步。根据设计要求，划分动作步，并画出相应的动作顺序简图。图 3-62a 本例的工作顺序简图。从图可看出它的工作过程及可能的分支。

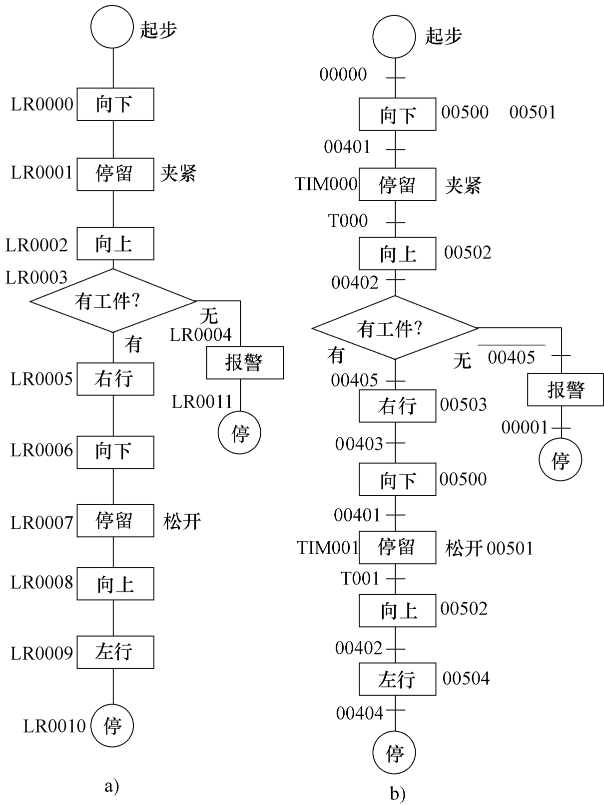


图 3-62 机械手动作过程简图

画动作顺序简图实际是设计要求的图形表示，弄清了设计要求，完成这一步是不难的。只是在这个图上，还要记上要使用的继电器地址。即如图所示的 LR000、LR001……

2）分配输入输出地址。进入各个步，都要完成一定的动作。动作与相应输出点的状态有关。为此，就要分配与动作对应的输出地址。本例的动作（输出）分配如下：

下降——00500 ON，下降；OFF，下降停止。

上升——00502 ON，上升；OFF，上升停止。

夹紧——是用弹簧实现的。0501 ON 弹簧松开，机械手松工件；0501 OFF 弹簧夹紧工件。右移——00503 ON 右移；OFF，右移停止。

左移——00504 ON 左移；OFF，左移停止。

步的转换是有条件的。而条件多为输入产生的。所以，也要为输入分配地址。本例的条件（输入地址）分配如下：

下限位开关——00401 ON，达下位；OFF 离开下位

上限位开关——00402 ON，达上位；OFF 离开上位

有工件——00405 ON，有工件夹住；OFF，无工件。

右限位——00403 ON，达右位；OFF，离右位。

左限位——00404 ON，达左位；OFF，离左位。

起动——00000 ON 起动。

3）流程图设计。过程图画出后，并作了 I/O 分配，弄清了步转换条件，则可在过程图的基础上，加入转换条件，并指定了相应的地址，即为流程图设计。图 3-62b 已设计好了本例的流程图。从图知：

起动后（00000 ON），即进入第一步。这时，00500 ON，00501 ON，可使机械手松开，并下降，准备抓取工件。

达到下位（00401 ON）后，使 00500、00501 OFF，下降停止，并夹紧工件。为可靠计，应令其延时 2s，可以保证夹住工件。为此，此时起动定时器 TIM000。定时器定时到，使机械手上升，即 00502 ON。

到达上限，即 00402 ON 后，上升停。判断机械手是否夹到工件。若无工件，00405 OFF，它的常闭触点（$\overline{00405}$），即 00405 的非为 ON，这时产生报警，机械手不再工作。这时，如使 00001 ON，可停止报警，并使步进程序结束。若有工件，00405 ON，则机械手右移（00503 ON）。

右移到右位，00403 ON，右移停止，并开始下降，即 00504 ON。

下降到下限位，00401 ON，00501 ON，松开。为可靠松开，也延时 2s。为此起动定时器 TIM001。

TIM001 时间到，则 00501 OFF，并使 00502 ON。前者机械手又夹紧，但已无工件，无关紧要。后者使机械手上升。

到上限位，00402 ON，上升停，并 00504 ON，左移。

到左限位，00404 ON，左移到，步进程序结束。又等待 00000 新命令，再起动新的过程。图 3-62b 与图 3-62a 不同的是，它的动作与条件是以 I/O 编号表示的，更详细一些。

4）建立步进逻辑程序。本例假设用的是 PLC 有 STEP 及 SNXT 两条步进指令，可用以

建立步进逻辑程序。这 8 步及辅助的步的程序如图 3-63 所示。

5）建立步与动作的逻辑关系。由于本例子较简单，图 3-63 也把这个关系画了出来。从图 3-62 知，它共有 9 步。即 STEP LR0000 ~ LR0008。另外还有两个停止步，即不带标号的 STEP，分划处在 SNXT LR0011 及 SNXT LR0010 之后。这意味着，只要执行这两条 SNXT，步进程序即行结束。

再者，这里的第四步，即STEP LR0003，有两个分支，00405 的 ON 或 OFF 区分。ON 则转为 STEP LR0005，机械手右移。而 00405 OFF 则转为 STEP LR0004，报警。若此时使 00001 ON，报警停，也使步进程序结束。

应指出的是，步进程序可与别的程序并存。如还有别的程序，PLC 除进行机械手控制之外，还可进行别的控制。

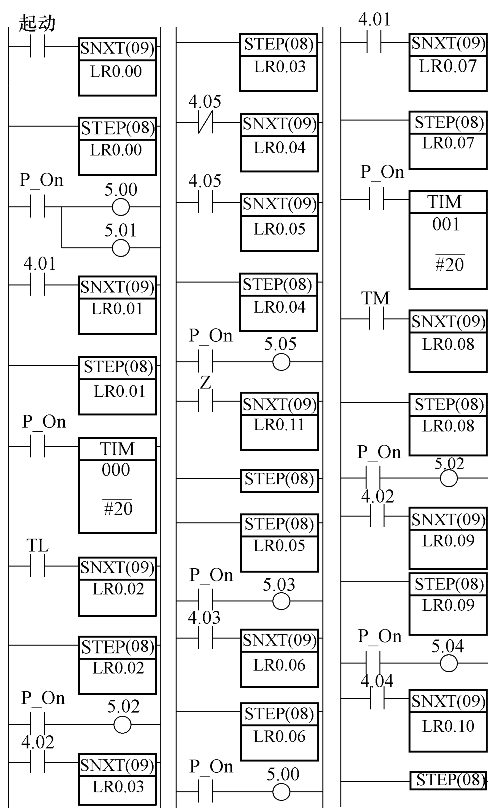


图 3-63 步进逻辑程序

附带说明一下 OMRON 步指令的特点：

① 步指令起动，可用任何一个步号，不一定非从第一位开始。只要使对应的继电器（如图 3-62 所示的 LR0000 ~ LR0002）ON 即可。

② 进入步后，起动位可不保持。步信号（上例的对 LR 继电器位）会自保持。故可用脉冲信号起动步。而且，即使起动信号保持，步的程序完成后，也不会再起动步程序。而要再起动，先要把起动信号复位（OFF），然后再 ON，才能再起动步工作。

③ 转入新步后，旧的步先复位，然后起动新步。新的步与旧的步不同时工作，但总有一个工作，这点与别的 PLC，如三菱的，略不同。后者，新旧可同时 ON 一个周期。旧的步复位后，其所有输出均复位，但计数器保持当时值。

④ 步指令使用的继电器不受限制，任何继电器均可用。只要不与别的重复使用即可。这说明，它的步数几乎是不受限制的。而别的厂家的 PLC 步的使用多是特殊指定的继电器，如三菱、西门子，用的是步继电器 S。

⑤ 步的程序可与一般程序并存。而且，在一个程序中，还可有多组步程序。只是它不能调子程序。

要注意的是，按 OMRON 步进指令的使用要点，这些步进指令应放在子程序之前，主程序之后。如把它放在子程序之后，这些步指令将不执行。如把主程序放在它之后，那放在它之后主程序将不执行。

再就是，这些“动作完成”信号，最好“脉冲化”，一旦进入新“动作”，相应的“动作完成”信号即可 OFF。这样，可避免进入新动作时，不会错误产生动作完成信号而出错。

用步进指令和流程图法进行程序设计是很方便的，只是稍繁一些，程序量大些。但扫描时间不增加，而且反而少了。因为未起动的步，或已复位的步，程序是不扫描的。

（2）组合机床动力头运动控制。设计要求见 3.3.4 节 3. 组合机床动力头运动控制。在此，用流程图法再对其进行设计。

1）划分动作步。根据动作要求，可分为六步，即第一次快进（LR0.01），第一次进（LR0.02），第一次快退（LR0.03），第二次快进（LR0.04），第二次进（LR0.05），第二次快退（LR0.06），如图3-64所示。

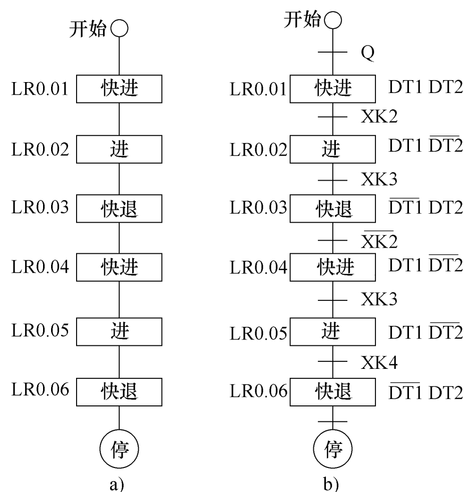


图 3-64　动力头运动控制动作步划分

2）分配输入输出地址。为了简化，全部用符号地址。

输出用 DT1（ON 进），$\overline{DT1}$（OFF 退）。DT2（ON 快），$\overline{DT2}$（OFF 慢）。为输出符号地址。

输入用 Q ON（起动）。XK1、XK2、XK3、XK4 为各相应的行程开关 ON。$\overline{XK2}$ 为 XK2 OFF。

3）设计流程图。根据题意，对每一步的输出及步到步之间的转换条件进行标注，如图 3-64b 所示。

4）建立步进逻辑程序。这里用步进指令建立步程序。共 6 步，分别用 LR0.01、LR0.02……代表这 6 个步。如图 3-63 所示。

5）建立步与动作的关系。按要求，在每步中，都用常 ON 触点（P_ON）使输出 DT1 或 DT2 ON。在进入新的步后，如不使其 ON 的，即自动使其转为 OFF。这是 OMRON 步进指令的特点。故这里使输出 OFF 的操作不必加入。具体梯形图如图 3-65 所示。

（3）煤气加热炉切阀、转换阀控制。设计对象的工作情况与工艺要求与本章 3.3.4 节 5. 煤气加热炉切阀、转换阀控制相同。输出也用 3 个变量 L、M、R，分别代表左煤气切阀，空气烟气转换阀及右煤气切阀方向控制。设计过程如下：

1）划分动作步。根据动作要求，可分为六步。即第一步 LR0.00、输出 100，第二步 LR0.01、输出 000，第三步 LR0.02、输出 010，第四步 LR0.03、输出 011，第五步 LR0.04、输出 010，第六步 LR0.05、输出 000。如图 3-66a 所示。

2）分配输入输出地址。为了简化，全部用符号地址。

输出用 L（ON 左切阀开，OFF 左切阀关），M（ON 转换阀置右，OFF 转换阀置左），R（ON右切阀开，OFF 右切阀关）。

输入用 A（主令信号），Ld、Ln、Md、Mn、Rd、Rn（反馈信号，含义同样 3.3.4 节 5. 煤气加热炉切阀、转换阀控制）。

3）设计流程图。根据题意，对每一步的输出及步到步之间的转换条件进行标注，如图 3-66b 所示。

4）建立步进逻辑程序。这里用步进指令建立步程序。共 6 步，分别用 LR0.00、

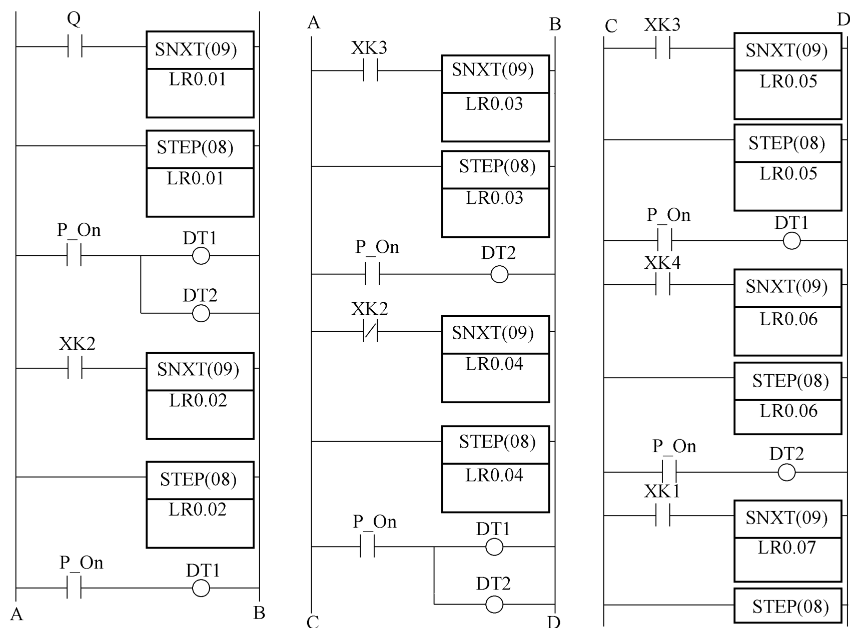


图 3-65　动力头运动步进控制梯形图程序

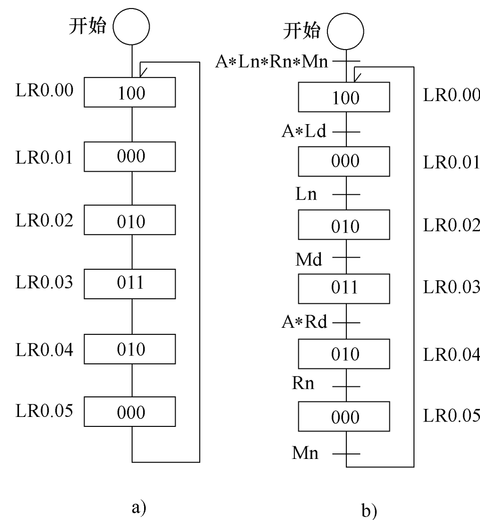


图 3-66　切阀、转换阀步进控制

LR0.02……代表这 6 个步。

5）建立步与动作的关系。按要求，在每步中，都用常 ON 触点（P_ON）使输出 L、M 及 R ON。在进入新的步后，如不使其 ON 的，即自动使其转为 OFF。这是 OMRON 步进指令的特点。故这里使输出 OFF 的操作不必加入。具体梯形图如图 3-67 所示。

从以上介绍可知，用流程图法，设计那些步骤分明的程序，是很简单的。

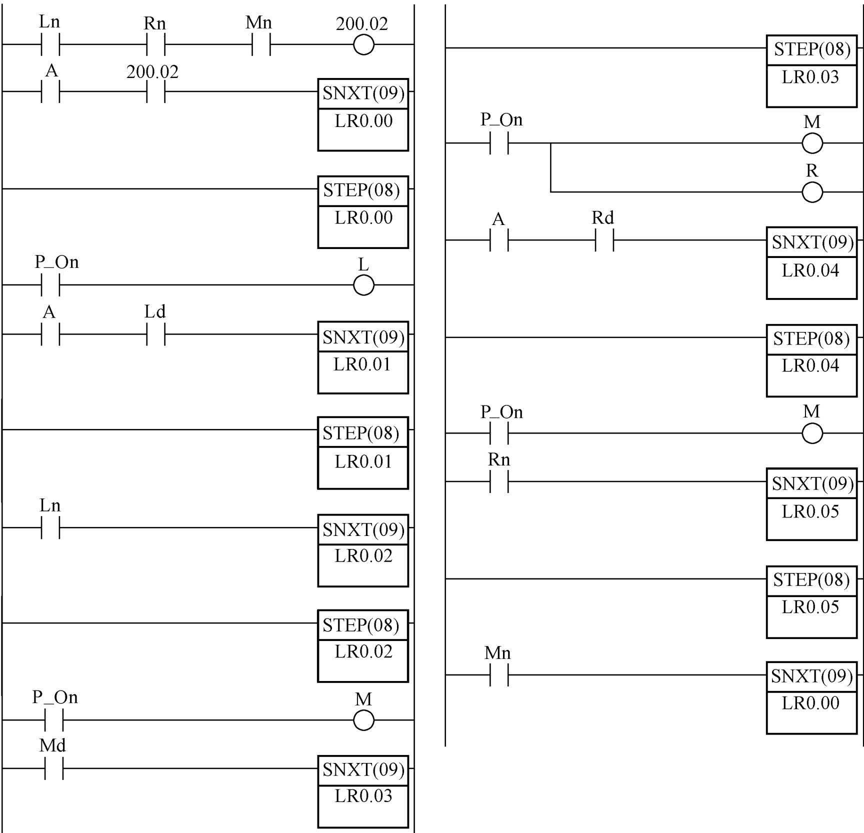


图 3-67　切阀、转换阀步进控制程序

应指出的是，流程图法多用于步进控制程序的设计，但不等于只能用于这种程序的设计。其实，条件逻辑控制也可用它，只要用相应的中间继电器作好“记忆”，处理好逻辑“相混”问题，不仅可用，而且用起来也很方便。

3.6　高级逻辑设计法编程

关键词：字逻辑、屏蔽、形式参数、实际参数、宏、功能块

以上讨论的逻辑问题都是以位为单位。控制一个输出，就要一组逻辑关系。其实，有时虽要控制很多输出，但其逻辑关系是相同的。这时，可否用“字”逻辑处理指令去代替“位”逻辑处理指令？或用子程序、宏去处理？

“字”逻辑设计也可说成集成化设计。往往用很少的字节或字或双字处理指令，取得用很多“位”处理指令的所具有的功能，可大大提高程序的效率。

3.6.1　用字逻辑指令处理

1. 双按钮多位起、保、停逻辑

两个按钮操作的起、保、停逻辑是，起动按钮与输出触点并联，然后再与停车按钮取反

后串联，以此作为输出线圈的输入条件。用字操作时，也是这个思路。但可以实现一个字（16 位）的起、保、停逻辑。

图 3-68 所示为字操作、起、保、停逻辑。0 通道的各个位接起动按钮，1 通道的各个位接停车按钮，10 通道的各个位接输出线圈。200 为中间继电器通道，以用作过渡。

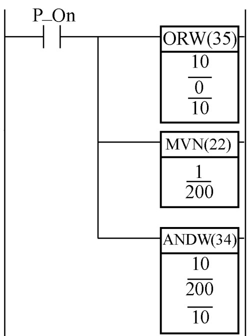


图 3-68 字起、保、停逻辑

图中 P_ ON 为常 ON 触点。说明这 3 条指令总是在执行。所以，0 通道任一位 ON，都可使 10 通道的相应位 ON，并将予以保持。而 1 通道的任一位 ON，则必使 10 通道的相应位 OFF。而且，这里也是停止优先。即起、停按钮同时按时，为停止。

图 3-68 用了 4 条指令，即可对 16 路进行起、保、停控制。而用基本逻辑指令，每条起、保、停就需 4 条指令，16 路起保停需 16 ×4 =64 条。4 条代替常规的 64 条，程序的效率当然高多了。

2. 单按钮多位起、保、停逻辑

一个按钮起、保、停逻辑也可用字操作。图 3-69a 所示的就是这个逻辑。这里 0 通道各个位为按钮输入，11 通道的各个位为工作输出。

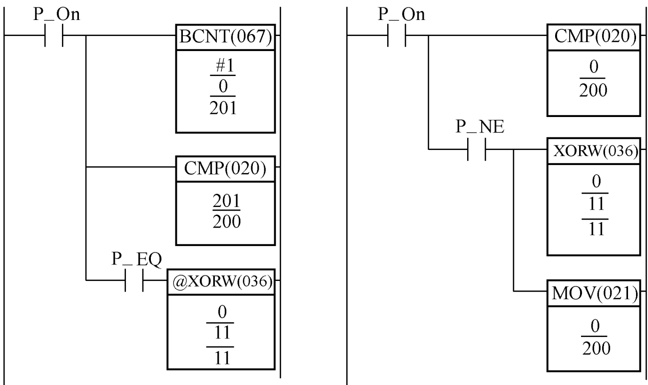


图 3-69 单按钮字起、保、停逻辑

图中 P_ON 为常 ON 触点，使统计在通道中置 ON 的位数 BCNT 及比较 CMP 指令，在每一扫描周期都得以执行。BCNT 指令有 3 个操作数，对本例，第一个操作数，#1，为要统计的通道个数为 1，第二个操作数为 0，要统计的开始通道为 0 通道，第三个操作数为 201，统计的结果存于 201 通道。

这时，若有一个按钮按下，统计结果的 201 通道内容变为 1，经比较（执行 CMP 指令），使相等标志 P_ EQ ON，进而使 0 的内容与 11 的内容进行异或逻辑操作，结果存于 11 中。

这个异或操作，将使 11 通道中，与按下按钮对应的位，改变状态，原为 OFF 的，将转为 ON，原为 ON 的，将转为 OFF，实现了单按钮起停。而其它位不变，因为 0 通道其它位为 0，0 与 1 或 0 异或，其结果不变。

该图 3-69b 也可起到此功能。而且“XORW”指令将自然为微分执行。

提示：这里的条件是，要确保在同一时间，16 个按钮只能一个按下，否则操作无效。在实际系统中，这个条件也可满足。

3. 非标准 I/O 分配处理

以上介绍的字逻辑操作都是针对 16 路的起、保、停逻辑，即标准 I/O 分配。但实际上可能不那么正好。如果少于 16 路怎么办？如图 3-70 所示的就是一种解决办法。

这里引入 200 及 202 通道作过渡。整个操作先对 200 进行。图中①是 MOV 指令，为对 200 操作做好准备。⑤是对 10（输出通道）的内容作屏蔽，仅保留不参与起、保、停逻辑的位，并存入 202 通道，以便于恢复这些位的内容。⑦为对 200 的内容作屏蔽。由于 H03 的内容为 H02 的反，故它保留下来的为要参与起、保、停的逻辑。⑧为或指令，正好把不参与及参与起、保、停逻辑的位，合成后赋予 10 通道。这样，既可保证参与的位，有起、保、停功能。而不参与的又不受其干扰。

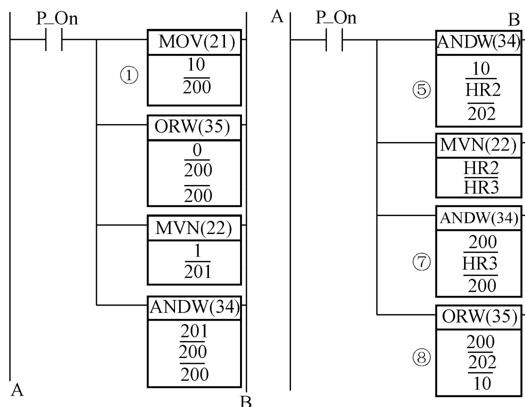


图 3-70　非标准 I/O 分配字起、保、停逻辑处理程序

H02 通道，其值可现设定，且可掉电保持；可使程序具有弹性。执行本程序，如 H02 的值设为 F，则 0、1 通道的相应位仅能对 10 通道的 04 ~ 15 位作起、保、停控制。10 通道的 00 ~ 03 位可接受其它控制，不受此程序影响。

如果仅是个别位另有所用，不一定要用图 3-70 的办法，也可先作字处理，在此之后，个别位再作处理。这时，尽管存在多输出，但仍可达到目的。因为，PLC 指令是一条条执行的，后执行的指令，可改变先执行指令的执行的结果。

4. 测试逻辑

一些自动控制系统在工作前，往往要不定时地对一些部件或指示部分作些测试。测灯逻辑就是一例。指示灯本来用以显示相应的信息的。若是坏了，就不能正确显示。为此，要用测灯逻辑对灯是否正常作测试。

图 3-71 为测灯逻辑。也是字操作。000 通道为测试输入，10 为测灯输出。可通过按下接于 0 通道的对应位的按钮，视相应灯是否亮，看灯是否正常。什么时候要测灯，使 2.00 ON 即可。这里，用了两条指令，即可进行 16 个灯的测试。程序的效率也是很高的。

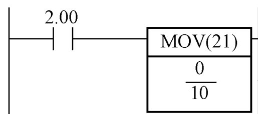


图 3-71　测灯逻辑

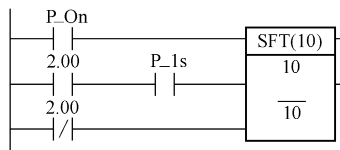


图 3-72　顺序测灯逻辑

图 3-72 所示的为用移位指令实现 16 个指示灯测试逻辑。2. 00 ON，开始测试。这里 P_1s为秒脉冲信号。每秒发一次脉冲，使 P_ ON（逻辑值为 1）信号移位给 10 通道的 00

位，而00位的状态移给01位，01的移给02位，等等。经历16s后，10的内容为FFFF，各位全为ON，可检查指示灯是否能正常工作。

检查完成后，可按2.00 OFF，其常闭触点，使SFT复位，同时，输脉冲输入切断。测试停止。

如果测试的灯多于16个，其终止通道可设得比10的编号大些。

5. 表决电路

图3-22介绍过3个开关表决电路。如表决的参加者较多，电路将很复杂。但用图3-73的逻辑，就比较简单。

该图程序用了BCNT指令，它检测从1通道开始，共1（这里第一操作数为常数1）个通道中，ON的位个数。结果存于200通道中。然后把200的内容与常数8比较，若有大于8，则产生输出，11.00 ON，表示表决已超过半数，通过。

如果参加表决的位数比16多，那BCNT指令的第一个操作数可设得比1大。如设为#2，则为32个位，#3为48个位，等等。这时，作比较的第二个操作数不应是8，也要作相应调整。

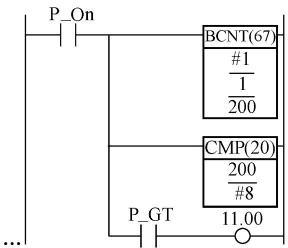


图3-73　表决电路

如果参加表决的位数比16少。可先对1作逻辑与运算，屏蔽掉不参与的位。比较指令中的常数不用8，再改为相应的数也就可以了。

6. 多个开关控制一个输出电路

图3-21介绍了3个钮子开关控制一个灯的逻辑。但要有多个钮子开关，如10个、20个，逻辑就相当复杂了，所用的触点数将是很多的，以至于出现“组合爆炸”。但，如用字或多字逻辑处理，就好办了。图3-74即为16个开关控制一个灯的逻辑。

从图知，它用0通道作16个开关的输入点，程序运行时，不断比较0通道与HR内容。如两者不等，则把0通道的内容传到HR0通道。同时，10.00改变状态。如两者相等，则10.00状态不变。显然，这个逻辑起到了所要求的用16个开关控制一个灯作用。

按此原理，如果为32个开关，再多作一次比较就是了。用多少开关可不受限制。

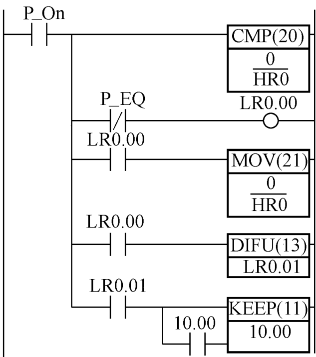


图3-74　多个钮子开关控制一个灯电路

3.6.2　用子程序处理

逻辑关系是相同的，还可用子程序、宏处理，而不必对每一个位都做雷同的编程。但有关地址要分配好。

OMRON PLC子程序无法带参数调用。但可作适当处理，达到带参数调用的目的。而且，也只有带参数调用子程序，才能在逻辑关系相同问题的处理时，简化编程。

1. 带参数调用子程序处理

图3-75b所示为子程序体。其使用的地址有200.00、200.01、200.02三个。200.00为输入信号，在程序中不产生输出。200.01与200.02产生输出，而且，其触点还作为输入的

一部分，起到反馈作用。

该子程序的功能是，200.00 ON、OFF 一次，200.01 ON，再 ON、OFF 一次，200.01 OFF。当然，这只是一个例子。其实，子程序的内容不同，可实现的功能也将不同。

显然，如不是带参数调用子程序，那只是对 200.01 的处理，对别的地址就不起作用。如要想对雷同的逻辑都用它处理，就要带参数调用它。

这里用的参数就是 200.00、200.01、200.02、200.00 为输入参数，200.01、200.02 既有输入，又有输出，为输入输出参数。因为 200.00、200.01、200.02 地址只是在子程序中引用，不是实际要处理的地址，故又称之为形式参数（地址）。

要带参数调用，就要在调用子程序前，先对形式参数赋值，把要处理的地址，即实际参数值赋给形式参数。在调子程序之后，则要做相反的赋值，把形式参数赋给实际参数。那样，即可用一个子程序，处理多个逻辑关系雷同，但地址不同问题。

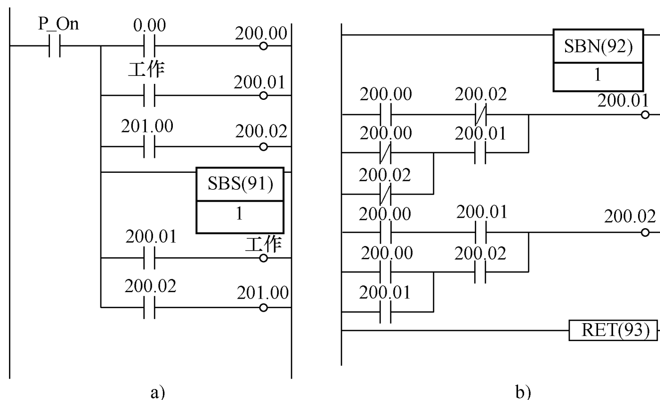


图 3-75　带参数调用子程序

图 3-75a 所示的就是一个带参数调用例子。图中 P_ON 为常为 ON 的触点，说明在它之后的所有指令，在每扫描周期都将执行。在这些指令中，先是 3 组简单的赋值逻辑，把 0.00、“工作”、201.00 的值（0 或 1）分别赋值给 200.00、200.01、200.02。其目的是把实际参数的值赋给形式参数，使调子程序时，将按 0.00、“工作”、201.00 实际值处理。

接着为调子程序。显然，如果执行后，200.00、200.01、200.02 的值将取决于调它之前的 0.00、“工作”、201.00 实际值。

调子程序指令之后，有两组简单的赋值逻辑，把 200.01 赋值给“工作”、200.02 赋值给 201.00。其目的是把形式参数的值赋给实际参数，完成调子程序，处理 0.00、“工作” 及 202.00 的目的。

为了简化调用时赋值的处理，也可在调子程序前指定一个存储参数的地址——指针地址。另外，再在子程序中，增加一些有关用这个指针地址，进行上述两种赋值的逻辑。

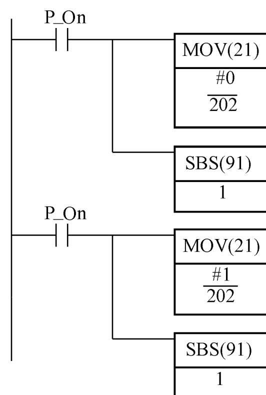


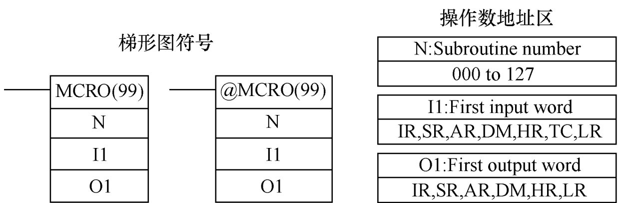
图 3-76　不同参数调子程序

图 3-76 是主程序，它调两次子程序，第一次调制前，先把常数 0 赋值给 202 通道（在子程序中指定为指针地址，如图 3-76

所示)，然后才调子程序 1。这时，希望用输入点 0. 00 去控制输出点 10. 00。第二次调制前，先把常数 1 赋值给 202 通道，然后才调子程序 1。这时，希望用输入点 0. 01 去控制输出点 10. 01，等等。由于在这两次调之前，对指针地址的赋值不同，其结果也将是不同的。在子程序中，怎样增加赋值处理，在此略。

2. 用宏处理

OMRON PLC 子程序无法带参数调用，但它有宏指令 MCRO（99)，也是用以调子程序，只是它为带参数的子程序调用。宏指令 MCRO 梯形图符号为



指令块中 99，为本指令的功能码。N 为将调用的子程序号。I1 为输入首地址，O1 输出首地址。这里的 I1、O1 即为形式参数。

I1 为输入，从通道 I1 开始有四个通道，I1、I1 +1、I1 +2、I1 +3，可做输入实际参数，从 O1 为输出，从通道 O1 开始也有四个通道，O1、O1 +1、O1 +2、O1 +3，可做输出实际参数。子程序中的对应地址为形式参数。只是不同机型，形式参数地址也不同。如 CPM 机与 I 对应的地址为 232 ~235，与 O 对应的为 236 ~239。CQM1 与 I 对应的地址为 96 ~ 99，与 O 对应的为 196 ~ 199。C200HS，E，G，X 机分别为 290 ~ 293 及 294 ~297。

至于可使用的子程序号，不同的机型也不同。对 CPM1 为 0 ~49，对 C200HS 为 0 到 99，对 CQM1、C200HX 等为 0 ~255。

当逻辑条件满足，执行本指令；否则不执行。本指令可正常执行；也可微分执行，即只在逻辑条件满足的第一周期执行。

本指令执行前，先把 I1 ~I4 的内容传送给子程序的形式参数，如为 CPM 机，232 ~235。然后执行子程序 N。

执行完子程序后，再把 235 ~239 的内容传送给 O1 ~O3。

图 3-77 所示出宏的调用简图。当然，子程序中形式参数地址没有使用时，相应的实参内容也不会改变。

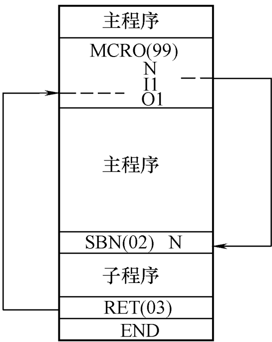


图 3-77 宏的调用简图

图 3-78 为运用宏指令的实例。图 3-78b 为子程序。它的输入只用一个位，232. 00。输出用了两个位，236. 00、236. 08。图 3-78a 为主程序，这里作了两次调用。

第一次调用时，实际参数为 0 通道及 10 通道，0. 00 对应于 232. 00，10. 00 对应于 236. 00，10. 08 对应于 236. 08。这可实现用 0. 00 对 10. 00 起停控制。

第二次调用时，实际参数为 1 通道及 11 通道，1. 00 对应于 232. 00，11. 00 对应于 236. 00，11. 08 对应于 236. 08。这可实现用 1. 00 对 11. 00 起停控制。

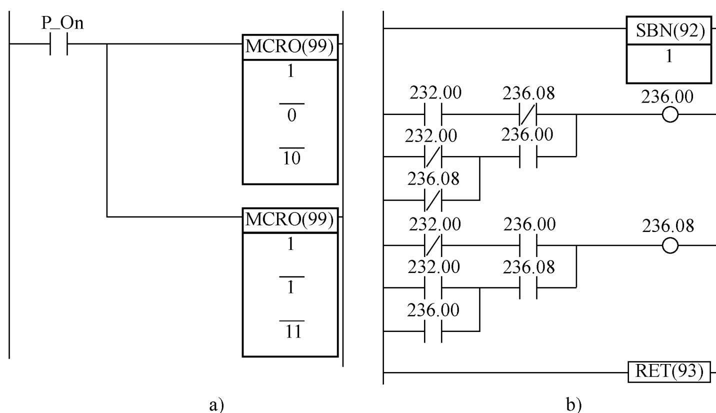


图 3-78　宏的调用

宏虽可带参数调用，调用较方便，但其结构是受限制的。其子程序只能是由触点的串、并构合，而且，其地址也要严格对应。否则不能用，或在用时要作适当处理。

3.6.3　用功能块处理

OMRON 新型 PLC 增加有功能块。功能块的编写及调用已在本书第 2 章作过介绍。在此，再介绍一个实例。

功能块名为“功能块 1”。其功能与图 3-29 单按钮起停程序相同。定义变量为，输入：II、oI、nI，BOOL 变量，输出：oO，nO，BOOL 变量。ST 语言程序如下：

oO：=（II AND nI = FALSE）OR（（II = FALSE OR nI = FALSE）AND oI）；

（＊ 图 3-29　单按钮起停电路条 0 的 ST 语言表达＊ ）

nO：=（II = FALSE AND oI）OR（（II OR oI）AND nI）；

（＊ 图 3-29　单按钮起停电路条 1 的 ST 语言表达＊ ）

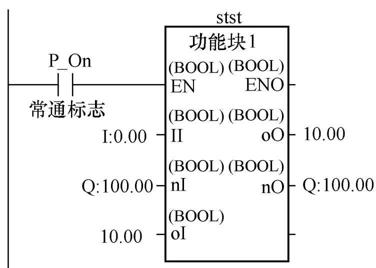


图 3-79　起保停功能块调用

“功能块 1”功能块的调用如图 3-79 所示。它用的实例名为“stst”。执行本调用，可实现 0.00 对输出 100.00 的起停控制。

3.7　标志逻辑设计法编程

关键词：记忆、比较、标志、设定值、实际值

以上讨论的逻辑处理比较精确，但都是基于与、或、非运算实现的，用的都是 PLC 的基本的逻辑处理指令，类似计算机用汇编语言编程那样，太“底层”了。其实，PLC 有很多功能很强的指令，完全可用它较简单地处理一些较复杂的逻辑问题。本节讨论的标志值法也许是其中一个较好的方法。

3.7.1 基本思路

基于与、或、非运算的逻辑处理，只是人们思考逻辑问题方法的一种数学抽象。它的优点是 PLC 的基本逻辑处理指令（与、或、非）就够用了。

其实人们思考问题用到的方法很多，其中一个最基本方法是“记忆”加“比较”。显然，人们努力学习，追求的不就是要能记更多的事，有更强的比较判断力，从而提高自身的思考力吗？相反，如果一个人没有记忆力、记的事情少，不会比较、没有什么判断能力，那这个人就如同婴儿，就不能思考任何问题的。

对人们这种“记忆”加“比较“的思考方法，是否也可加以抽象，作为 PLC 逻辑处理的一种算法呢？

答案是肯定的，这就是这里即将介绍的标志值法。

标志值法基本思路有两点：

“记忆”——设定好并记住标志的设置值，同时，不断监视标志的实际值。

“比较”——对标志的实际值与标志的设置值定期地进行比较，并依不同的比较结果产生相应的控制输出。

由于 PLC 有很丰富的、与这个“记忆”、“比较“相对应的指令，所以，实现这个算法是不难的。而且，这种算法更接近人们的思维方法，类似于用高级语言编程一样，人们更易理解。

3.7.2 实现方法

“记忆”的实现方法：

最常用的办法是用传送指令、MOV，用它传送标志的设定值，用它传送与输入信号对应的标志实际值。此外，也可用计数器计标定实际值。当然，其它数据处理指令，如算术运算、数据转换等指令，也可用。

“比较”的实现方法：

最常用的办法是用基本的比较指令、CMP，用它对标志值与预期值进行比较，依不同的比较结果（大、大等、等、小等、小）产生不同的控制输出。由于 PLC 技术的发展，它的指令系统越来越丰富。目前多数 PLC，除了这个基本的比较指令外，还有表比较、范围比较等功能更强的比较指令。这类指令可设定很多预期值，比较后可得到很多不同的结果。

3.7.3 实际应用

1. 起、保、停逻辑

传统的起、保、停逻辑都是用与、或、非的算法实现的，很简单。其实，也可用标志值法实现。只是这么简单的逻辑问题，没有必要用它就是了。但为了读者能具体地了解标志值法，以下介绍两个简单起保停逻辑例子。

（1）两个按钮实现起停

如图 3-80 所示，输出触点为 10.05。输入有两个按钮，接 202.00、202.01。标志字为 LR0。预期值为 1。标志值两个，0 与 1。从图可知，202.00 ON、202.01 OFF，则 LR0 的内容为 1。P_ON 为常 ON 触点，故每扫描周期都要执行比较指令。经比较，相等标志 P_EQ

ON，故 10.05 ON，实现起动。这之后即使 202.00 OFF，LR0 无新数传入，仍保持为 1，故输出可保持。而 202.00 OFF、202.01 ON，则 LR0 的内容为 0，经比较 P_EQ OFF，故 10.05 OFF，实现停止。这之后即使 202.01 OFF，LR0 无新数传入，仍保持为 0，故输出可仍停止。如 202.00、202.01 全 ON，由与 202.01 ON 后起作用，LR0 为 0，故仍停止，也是停止优先。

可见，它完全可实现基于与、或、非的起、保、停算法的。

（2）一个按钮实现起停

如图 3-81 所示，输出触点为 10.06。输入有一个按钮，接 202.05。标志字为计数器，CNT 001。预期值为 1。标志值为 CNT 的计数值是 2 与 1。从图可知，202.05 ON 一次，则 CNT 001 的内容减 1。减到 0，计数器将复位，其内容又变为设定值 2。

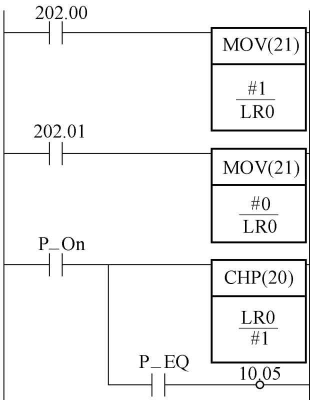


图 3-80　两个按钮实现起、保、停梯形图程序

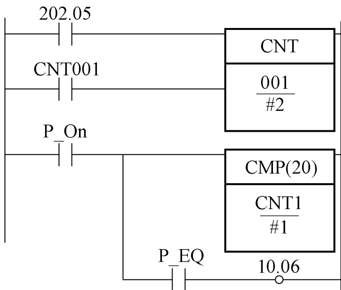


图 3-81　单按钮实现起、保、停梯形图程序

如 CNT 001 初值为设定值 2，减 1 后即为 1，经比较 P_EQ ON，故 10.06 ON，实现起动。这时再按按钮，CNT 001，再减 1，为 0，复位，其内容又变为设定值 2。这时，经比较 P_EQ OFF，故 10.06 OFF，实现停止。

再按按钮，又重复以上循环。可见，它完全可实现基于与、或、非的一个按钮实现起、保、停算法的。

2. 电梯电路之一

本章 3.3.4 节 2. 小车运动控制与电梯电路类似。在该例中，小车仅 5 个位置，但逻辑关系就已相当复杂了。若有 10 个、20 个位置怎么办？不仅关系复杂，而且指令将以阶乘关系增长。这将出现所谓“组合爆炸”（Combine Explosion）。

这类控制的顺序是非确定。到底向上或向下，依其所处位置及要前往的位置随机确定。处理这类问题有两种办法：

一是考虑所有可能，逐一列出它的逻辑关系，再确定其输出。可能性不多时，用这个办法是可行的，不少也是这么处理的。本章 3.3.4 节 2. 小车运动控制用的也是这个办法。

二是置标志（“记忆”），再判标志（“判断”），以确定输出。如电梯，电梯的工作总是处在停、上升、下降三个状态之一。其所处的位置可置个标志数（如层数），要去的位置也置个层数标志。这可用传送指令实现。判标志，则可用比较指令，如要求去的比现处的大，则向上，否则向下。相等则停。它是从 3 个可能的输出中按条件选取其中一个。

这么处理后，非确定顺序的问题，也成了有确定的处理步骤的问题。即随机控制确定化了。这比仅就逻辑条件的可能去组合，要简单得多。

过去，用继电器实现控制，则只能用第一种办法，靠逻辑条件的组合去实现。而 PLC 则不同，可用数字量作标志，可对数字量作比较，所以，可采用第二种办法，对此类问题进行处理，也才有可能。

图 3-82 就是用标志值法设计的，本章 3. 3. 4 节 2. 小车运动控制已讨论过的小车控制逻辑。如图所示，它按顺序给每一选择按钮指定一个编号，如 PS1（即输入点 0. 06）为#1，PS2（0. 07）为#2……也按顺序，对应地给每一行程开关指定一个编号，如 LS1（即输入点 0. 01）为#1，LS2（0. 02）为#2……哪个按钮 ON 或哪个开关 ON，就通过传送指令，把这个编号作为标志值，传送到 201 或 200 通道。

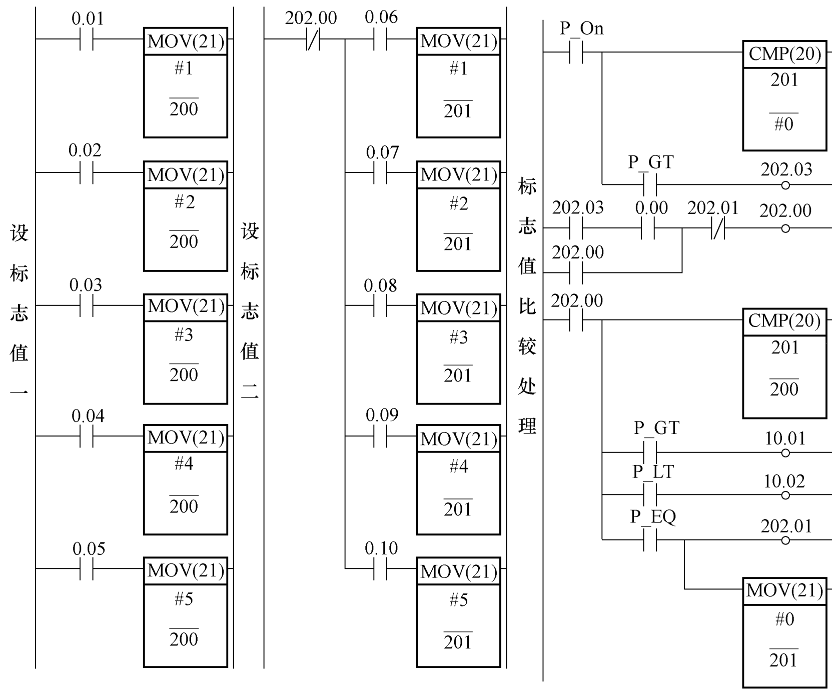


图 3-82　小车控制梯形图程序之一

执行传送指令之后，按起动按钮（0. 00 ON）。如选择标志值不为#0（说明已作了选择），则 202. 00 ON，并自保持。202. 00 ON，比较指令执行，比较 200 与 201 通道内容。如果 201 通道存的数比 200 的大，说明行程开关 ON 的编号比按钮 ON 的编号小，则比较大标志 P_GT ON，进而 10. 00 ON，使小车向右运动。

运动过程中 10 的内容将随行程开关动作而变化。当 200 的值变到与 201 的值相等，即达到所要求的位置，则比较相等标志 P_EQ ON，进而 200. 01 ON。这将使 202. 00 OFF，10. 01 OFF，运动停止。同时，用#0 传送给 201 通道，为新的选择作了准备。

如果 201 通道的数比 200 的小，即与上述情况相反。

把 202. 00 常闭触点串入向 201 传数的诸逻辑梯形图中，目的是一旦小车起动，就不再接受选择按钮送来的命令。待执行完原给的命令后，即小车停止运动后，才可接受新的

命令。

3. 电梯电路之二

以上用了传送指令“记忆”，这里用 DMPX 指令“记忆”，再对上例作进一步讨论。

DMPX 与 MLPX 配对的译码指令，是很常用的 PLC 指令。几乎所有厂家的 PLC 都有这两条指令。用它比用 MOV、CMP 指令有时效果更好。

图 3-83 示的就是用 DMPX 作标志值设置（“记忆”）。它可控制 15 个位置（对电梯讲就是 15 层）。通道 0 用以记录电梯实际所处层号。它的 01 ~ 15 位，对应第 1 ~ 15 层。通道 1 用以设要求到的层号，它的 01 ~ 15 位，对应要求到第 1 ~ 15 层。

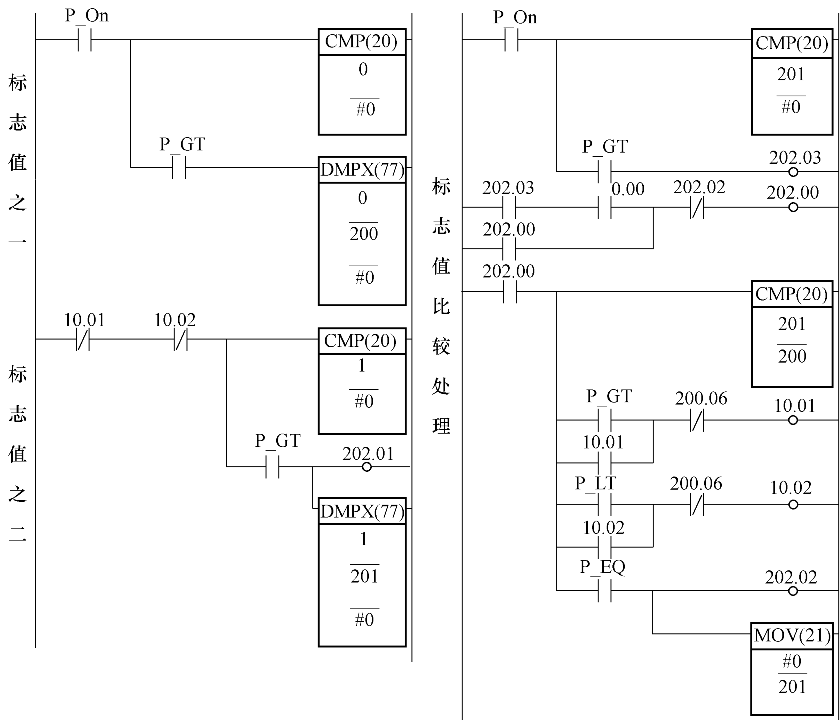


图 3-83　小车控制逻辑之二

从图知，它的标志值比较处理与图 3-82 完全相同。但标志值设定大为简化了。通道 0 或 1 的内容原为十六进制数，经 DMPX 译码后，得出的为通道中那一最高位 ON。对应的就是值 1 ~ 15。这正好就是图 3-82 要设的标志值。

可知，图 3-83 比图 3-82 简单，而它控制的功能却比后者强。OMRON 新型的 PLC 的 DMPX 指令，可实现 255 位的译码。用它可实现 255 层的电梯控制。即使世界上最高的建筑，也足够用了。

3.8　工程设计法编程

关键词：分散控制、分支控制、集中控制、配方控制、混合控制实现、分支混合控〖

3.8.1 分散控制及其应用

1. 分散控制程序要点

在本书第 2 章第 4 节，讲到动作控制及步进程序。本章第 5 节讲到流程图法编程。实际也就是分散控制。其控制可用基本指令实现，也可用移位指令实现，还可用步指令实现。“动作”、“步”可以是实际输出点，直接实现控制。也可不是实际输出点，间接实现控制。“动作完成”可以是实际输入点，直接实现反馈。也可不是实际输入点，间接实现反馈。

间接实现虽然要做些输入、输出的转换，但程序灵活、通用性强、易读、易改。是值得提倡的设计。

使用间接实现，所要作的工作有两个：

（1）根据“动作”或“步”的数量，设计与其相等的“动作”或“步”的控制程序。

（2）设计输入、输出转换程序。

分散控制也可能有分支，有平行分支与选择分支，以至于更为复杂的分支。图 3-84 为平行分支的原理图。

从图 3-84 知，它的“动作 2 完成“信号将起动两个动作，“动作 3”及“动作 33”。起动后，这两个分支将平行工作。直到这里的“动作 4 完成”、“动作 44 完成”信号都产生了，才能进入“动作 5”。

图 3-85 为选择分支。从图知，它的动作 2 完成后，有两个动作完成信号选择：“动作 2 完成 A”与“动作 2 完成 B”。如得到是“动作 2 完成 A”，则进入分支 A，直到分支 A 完成。如得到是“动作 2 完成 B”，则进入分支 B，直到分支 B 完成。

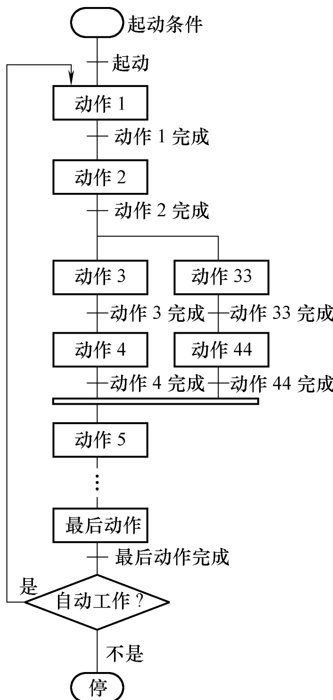


图 3-84 平行分支框图

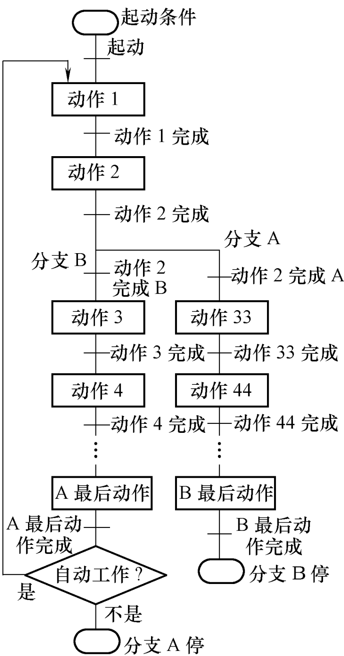


图 3-85 选择分支框图

2. 分散控制应用实例

（1）采用的是本章 3. 3. 4 节 3. 组合机床动力头运动控制，要求与其完全一样。

选用控制程序结合本例，用 6 个“动作”，具体程序略。按要求，其输出与“动作”的关系为

“动作 1”、“动作 2”、“动作 4”、“动作 5”DT1 应为 ON，动力头前进。其它情况为 OFF，动力头后退。

“动作 1”、“动作 3”、“动作 4”、“动作 6”DT1 应为 ON，动力头快速。其它情况为 OFF，动力头慢速。

其逻辑式应为

DT1 =“动作1”+“动作2”+“动作4”+“动作5”

DT2 =“动作1”+“动作3”+“动作4”+“动作6”

按要求，其输入与“动作完成”的关系为

2XK ON 应产生与“动作 1 完成”信号；3XK ON 应产生与“动作 2 完成”信号；2XK OFF 应产生与“动作 3 完成”信号；3XK 再次 ON 应产生与“动作 4 完成”信号；4XK ON 应产生与“动作 5 完成”信号；1XK ON 应产生与“动作 6 完成”信号。

所以，其逻辑式应为

“动作1完成”=2XK

“动作2完成”=3XK

“动作3完成”=2XK 非

“动作4完成”=3XK

“动作5完成”=4XK

“动作6完成”=1XK

“原位”=1XK

图 3-86 所示为相应的输入、输出逻辑梯形图。

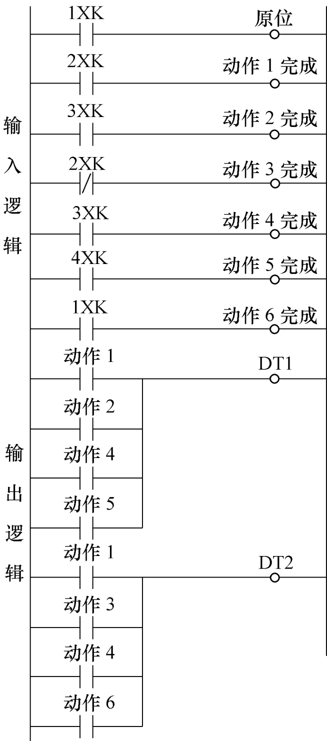


图3-86　输入、输出逻辑梯形图

（2）要设计一个功能与本章 3. 5. 1 节 2. 时序图法设计实例相同的梯形图程序。

结合本例，用 6 个“动作”，具体程序略。用的输出地址与原定的相同，也是 10. 02、10. 03、10. 04 三个。

其输出与“动作”的关系为

“动作 1”、“动作 4”、“动作 5”10. 02 应为 ON，喷头 A 工作。其它情况为 OFF，喷头不工作。

“动作 2”、“动作 4”、“动作 5”10. 03 应为 ON，喷头 B 工作。其它情况为 OFF，喷头不工作。

“动作 2”、“动作 3”、“动作 5”10. 04 应为 ON，喷头 C 工作。其它情况为 OFF，喷头不工作。

其逻辑式应为

10. 02 =“动作1”+“动作4”+“动作5”

10. 03 =“动作2”+“动作4”+“动作5”

10. 04 =“动作2”+“动作3”+“动作5”

按要求，其动作完成为定时信号。所以，应用定时器控制。共用6个定时器，TIM001、TIM002、TIM003、TIM004、TIM005、TIM006。分别与“动作1完成”、“动作2完成”、“动作3完成”、“动作4完成”、“动作5完成”、“动作6完成”对应。定时器时间设定，按要求分别为5、5、5、2、5、3s。对应的定时器设定值为#50、#50、#50、#20、#50、#30。

与这个输入、输出逻辑对应的梯形图程序如图3-87所示。

提示：这里的反馈用的都是事件信号，所以，本例用一下讲的集中控制更好。

总之，分散控制是很常见的控制算法。只要用这里的“动作”去控制对应输出点，再用对应的输入点去控制这里的“动作完成”，就可实现如机床刀架运动那样部件的自动控制。

动作控制也可插入定时控制。如用某“动作”去控制某输出点的同时，还去起动一个定时器，令其计时，再用该定时器计时到信号作为该动作完成信号，则这一动作就是定时控制。这种逻辑也是用得很多的。

3.8.2 集中控制及其应用

1. 集中控制实现程序

图3-88所示为一个集中控制梯形图程序。

图3-87 输入、输出逻辑对应的梯形图程序

图3-88 集中控制梯形图程序

从图知，它由工作控制、集中控制器及虚拟输出逻辑几部分组成。这里的梯形图用的为符号地址。

集中控制器主要由两个增计数器及相应的存储区组成。OMRON PLC 小机型没有增计数器，故用可逆计数器 CNTR 000、CNTR 001，但减计数不用。计数器 CNTR 000 用以步进，而 CNTR 001 用于计时。

当系统起动，进入工作，计数器 CNTR 000 按每次 CNTR 001 计时的情况，作增计数。CNTR 000 计到设定值，再计入 1，其计数值自动回到 0，并产生输出（CNT 000 常开触点 ON，常闭触点 OFF）。这时，如自动工作 OFF，则其常闭触点将使工作线圈 OFF，工作停止；否则，又从 0 开始，又执行第一步动作。计数的设定值为间接数，取决于“总步数”的值。

提示：OMRON PLC 的“总步数”应为实际步数减 1。

计数器 CNTR 001 对 100ms 脉冲作增计数。处工作状态时，开始计，每 100ms 加 1。到了设定值，再加 1，即回到 0，并产生输出。复位后，又作为新的计数准备；产生输出也为 CNTR 000 计数器提供步进信号。

间接地址为 DM998，即以它的值为地址的 DM 单元的内容，作 CNTR 001 的设定值。而 DM998 的值为 DM996 的值加 CNT 000 的现值。这意味着这个设定值放在 DM 区的开始位置由 DM996 确定。

这里的输出是虚拟的，实际输出将由实际地址用输出逻辑确定。虚拟输出也用了间接地址。具体情况是：间接地址为 DM999，即以它的值为地址的 DM 单元的内容，作为“虚拟输出”值。而 DM999 的值为 DM995 的值加 CNT 000 的现值。这意味着这个“虚拟输出”在 DM 区的开始位置由 DM995 确定。

这里虚拟输出用了一个字，16 位。可对 16 个逻辑量进行控制。控制步数的变化对程序没有影响，步数多少只受 CNTR 000 最大值及数据区大小的限制。所以，这个程序的功效比用分散方法进行定时控制要强得多。只是在实际运行前，需对有关 DM 区作好设定。

程序工作过程：当“起动”信号 ON，“工作”输出将 ON，并自保持，系统进入工作状态。“虚拟输出”将从 * DM999 传来数据，将根据前者的内容产生虚拟输出（如要产生实际输出，可把此输出再作传递）。与此同时，CNTR 001 开始计数，每 100ms 加 1。

当 CNTR 001 计数到 * DM998 设定的值，再计入 1，其输出 ON，产生“步进”信号。从而使 CNTR 000 加 1 计数，DM998 也随之赋以新值（加 1），实现了步进，其虚拟输出则是新一步的设定值。

这样延续，直到 CNTR 000 计到“总步数”，再计入 1，其输出 ON，并自身复位（现值回到 0）。这时，如“自动”ON，则开始新的循环，继续工作；如“自动”OFF，“工作”OFF，“虚拟输出”置 0，系统工作停止。

提示：集中控制没有反馈，所以没有分支。

2. 配方控制

图 3-88 程序，还可增加配方控制。如图 3-89 所示，它的输出逻辑部分增加了新的控制指针，为 DM1999。它所指向的地址开始地址分别由 DM1995 确定。

用这新增加的这指针，取得与步输出对应的参数，如某某设定值。以在实施开关量控制

的同时，也对模拟量作控制。

当然，还可再增加指针，以取得更多的，与步输出对应的参数，即所谓“配方”，再用此对水泥搅拌生产进行控制是很方便的。

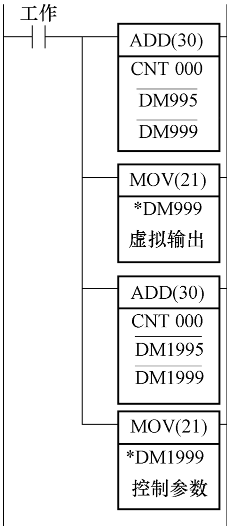


图 3-89 指针控制

3. 集中控制应用实例

（1）设计要求同本章 3. 5. 1 节 2. 时序图法设计实例的喷泉控制程序。

本例用图 3-88 集中控制的梯形图程序。数据设定分别是：

对 OMRON PLC：设其“虚拟输出”实际地址为 220。本例有 6 个工作步，故“总步数”设为 #5。

DM996、DM995，可任意设，只要所设的数据不被覆盖即可。本例 DM996、DM995 分别设为 #100 与 #0。即虚拟输出设定值地址，从 DM0000 开始；步的定时值设定值地址，从 DM0100 开始。

DM0000 ~ DM0005 依各步要求的虚拟输出设定。

DM0100 ~ DM0105 依各步要求的定时值设定。

有关这些 DM 区的设定值及其备注，见表 3-21。

表 3-21 参数选择

变量地址	变量值	备注
DM0000	#2	220. 01 ON
DM0001	#C	220. 02，220. 03 ON
DM0002	#8	220. 03 ON
DM0003	#6	220. 01，220. 02 ON
DM0004	#E	220. 01，220. 02，220. 03 ON
DM0005	#0	全 OFF
DM0100	#49	第 1 步定时值减 1
DM0101	#49	第 2 步定时值减 1
DM0102	#49	第 3 步定时值减 1
DM0103	#19	第 4 步定时值减 1
DM0104	#49	第 5 步定时值减 1
DM0105	#29	第 6 步定时值减 1
DM995	#0	
DM996	#100	

除了设定参数，还要设计实际输出逻辑，如果实际输出为符号地址，分别是：“AA 工作”、“BB 工作”及“CC 工作”。其实际程序如图 3-90 所示。

作了以上设定及加入图 3-88 实际输出逻辑后，再运行图 3-90 梯形图程序，完全可实现

所要求的功能。

（2）设计一个十字路口交通岗上的红绿灯控制程序。

其要求是，按预定的时序，控制十字路口红绿灯哪个亮，哪个不亮，不考虑行车、行人的情况。共有 6 个灯。南北向红、黄、绿，用 R1、Y1、C1 代表，东西向用 R2、Y2、C2 代表。其实际地址略。这 6 个灯能依时间变化工作，如图 3-91 所示。图中 s 为秒。

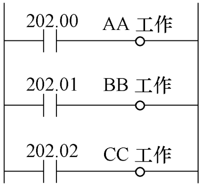


图 3-90　喷泉控制输出梯形图程序

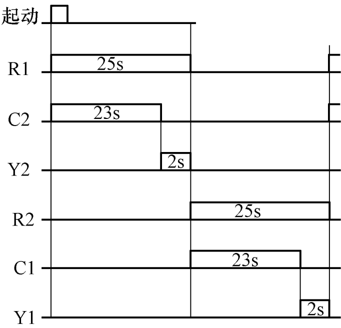


图 3-91　动作时序图

考虑到此系统为定时工作，故使用集中原则控制，用图 3-88 的梯形图程序。数据设定分别是：

本例有 4 个工作步，故“总步数”设为 #3。

DM996、DM995，可任意设，只要所设的数据不被覆盖即可。本例 DM996、DM995 分别设为 #100 与 #0。即虚拟输出设定值地址，从 DM0000 开始；步的定时值设定值地址，从 DM0100 开始。

DM0000 ~ DM0005 依各步要求的虚拟输出设定。

DM0100 ~ DM0105 依各步要求的定时值设定。

有关这些 DM 区的设定值及其备注，见表 3-22。

表 3-22　参数选择

变 量 地 址	变 量 值	备 注
DM0000	#22	220.01,220.05 ON
DM0001	#42	220.01,220.06 ON
DM0002	#14	220.04,220.02 ON
DM0003	#18	220.04,220.03 ON
DM0100	#229	第 1 步定时值减 1
DM0101	#19	第 2 步定时值减 1
DM0102	#229	第 3 步定时值减 1
DM0103	#19	第 4 步定时值减 1
DM995	#0	
DM996	#100	

实际输出逻辑，如图 3-92 所示。

作了以上设定后，再运行图 3-88、图 3-92 梯形图程序，完全可实现所要求的功能。

集中控制程序是很万能的，所控制的点数、步数及有关参数设定几乎都不受限制。惟一的限制是 DM 区的容量及实际输入输出点数。

附带在此提及的是，各 PLC 厂家多提供有凸轮控制器。如把增量式编码器与它连接，即可灵活地处理位置或时间相关任务。其实质与这里介绍的集中控制的机理基本是相同的。但它用模块实现，可以减轻 PLC CPU 负荷。而这里则是用程序实现。

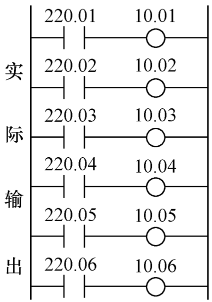


图 3-92　输出梯形图

3. 8. 3　混合控制及其应用

1. 基本混合控制梯形图程序实现

图 3-93 所示为混合原则控制梯形图程序。

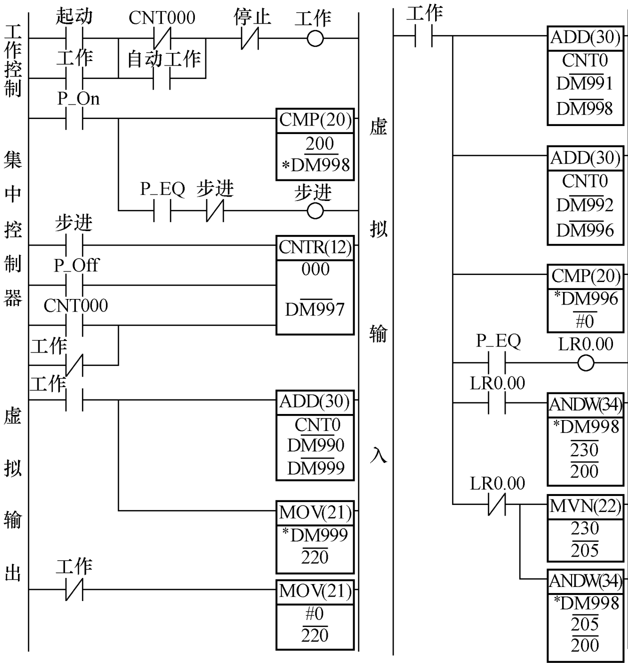


图 3-93　混合控制逻辑梯形图程序

从图 3-93 知，它由工作控制、集中控制器、虚拟输出及虚拟输入几部分组成：

（1）集中控制器。使用一个可逆计数器（CNTR 000，减计数不用）。当“工作”ON 时，每次“步进”ON，则 CNT 000 加 1，实现步进。计数到设定值（存于符号地址“总步数”中）后，再加 1，CNTR 000 现值又回到 0，同时，其常开触点 ON，常闭触点 OFF。这时，如自动工作 OFF，其常闭触点将使“工作”OFF，工作停止；否则，又将从 0 开始计数。而“步进”什么时候 ON，取决于“计算输入”通道的内容与 DM998 值指向的 DM 地址的内容进行比较的结果。当这两者相等时，即得到了应有的反馈信号，表示动作完成，则

“步进”ON。

（2）虚拟输入。用“虚拟输入”及“反虚拟输入”两个通道。两者内容相反，其对应位，如前者为1，则后者为0；如前者为0，则后者为1。而使用其中哪一位作为虚拟输入，由*DM998确定。*DM998的哪一位设为1，即使用哪一位作为反馈输入。而这个反馈输入是用正（ON），还是用反（OFF）信号，则取决于*DM996与*DM998的对应的位是怎么设的。设为1，反馈输入用的是反虚拟输入（用OFF信号）；设为0反馈输入用的是正虚拟输入（用ON信号）。

为此，在该图的程序中，要先进行* DM996与#0比较，如相等，则使用ON信号；反之，使用OFF信号。使用ON信号时，“计算输入”为*DM998直接与“虚拟输入”通道的内容作“与”运算；用OFF信号，“计算输入”为*DM998与“反虚拟输入”通道的内容作“与”运算。这个“计算输入”与*DM998比较，如相等，即收到应有的反馈，从而产生“步进”信号，并将引起计数器CTRN 000加1、步进。

DM998的值等于CNT 000现值与DM991之和，所以，DM 991决定了指针DM998的初值。DM996值等于CNT 000现值与DM992之和，所以，DM 992决定了指针DM996的初值。

（3）虚拟输出。使用“虚拟输出”通道。其值是由以DM999值为地址DM字的内容传来的。这个DM字的内容设成什么样，“虚拟输出”就有什么样的输出。DM999的值为DM995的值加CNT 000的现值。故DM999的初值由DM990内容确定。

（4）程序工作过程。当“起动”信号ON，“工作”输出将ON，并自保持，系统进入工作状态。“虚拟输出”将从*DM999传来数据，将根据前者的内容产生虚拟输出，如要产生实际输出，可把此输出再作传递。

随着实际输出控制的推进，系统的实际输入将传递给“虚拟输入”（有关程序另附）。程序将根据*DM998、*DM996的设定，把“虚拟输入”进行逻辑处理，然后得到“计算输入”。再把“计算输入”与*DM998的设定进行比较。直到两者相等，说明已完成此步控制，进而产生步进信号，使CNTR 000加1计数。DM998也随之赋以新值（加1），实现了步进，其虚拟输出则是新一步的设定值。

这样延续，直到CNTR 000计到“总步数”，再计入1，其输出ON，并自身复位（现值回到0）。这时，如“自动”ON，则开始新的循环，继续工作；如“自动”OFF，“工作”OFF，“虚拟输出”置0，系统工作停止。

有了本程序这个框架，在实际运行前，根据实际情况，对有关DM作好设定，同时，再增加必要的输入、输出程序，完全可实现相当复杂的顺序控制。以下用两个实例介绍怎么应用本程序。

2. 基本混合控制ST语言程序实现

混合控制逻辑梯形图程序也可使用ST语言表达。但要先设计功能块，然后再调用。本例功能块名为“功能块3”。

（1）变量声明

1）输入变量

```
stA（起动），        BOOL    0    FALSE    INPUT  0
stO（停止）  ，      BOOL    0    FALSE    INPUT  0
zdW （自动工作），   BOOL    0    FALSE    INPUT  0
```

wI　（工作），　BOOL　0　FALSE　INPUT　0
iW　（虚拟输出），　WORD　0　0　INPUT　0
iJS　（计数器），　UINT　0　0　INPUT　0
sJS　（总步数），　UINT　0　0　INPUT　0
2）输出变量
wO（工作），　BOOL　0　FALSE　OUTPUT　0
oW（虚拟输出），　WORD　0　0　OUTPUT　0
oJS（计数器），　UINT　0　0　OUTPUT　0
3）内部变量
inW1，WORD　100（数组大小）D1000（AT 地址）INTERNAL　0
onW，WORD　100（数组大小）D1100（AT 地址）INTERNAL　0
inW2，WORD　100（数组大小）D1200（AT 地址）INTERNAL　0
jnW（计算输入），　WORD　0　0　INTERNAL　0

提示：这里 wI 与 wO、iW 与 oW、iJS 与 oJS 实际是一个变量。新版 ST 语言可声明输入输出变量。用这些变量即可用同一变量名。

声明内部变量时，可在“编辑变量”窗口上，点击“高级”项。之后，将弹出“高级设置”窗口。如图 3-94 所示。

在高级设置窗口上，如图所示，选择 inW2 为数组变量，数组大小为 100，AT 地址为 D1200，即数组中的 inW2［0］即为 D1200，inW2［1］即为 D1201……这样，inW2 数组虽为内部变量，实际是 PLC 的实际地址。

本例声明了 3 个数组，数组大小都是 100，都是 AT 变量。inW1 为设定哪个位输入有效，inW2 为设定输入有效位用常开触点（设为 0），还是用常闭触点（设为 1）。onW 为设定输出。



图 3-94　内部变量高级设置窗口

（2）ST 语言程序

以下程序有注解。可参阅注解阅读。

```
wO：=(stA OR wI)AND  stO = FALSE;(＊ 工作起动停车逻辑＊ )
IF wI THEN(＊ 如果工作,执行以下语句＊ )
  oW：= onW[iJS];(＊ 把设定输出赋值给虚拟输出＊ )
  jnW：= inW2[oJS] XOR iW;(＊ 实际输入转换为计算输入,可反映是否用常闭触点输入＊ )
  jnW：= jnW AND inW1[oJS];(＊ 指定有效的输入位＊ )
  IF jnW = inW1[iJS] THEN
    oJS：= iJS + 1;(＊ 如果设定输入与计算输入相等,计数器加 1,转下一步＊ )
    IF oJS > = sJS THEN(＊ 如果已完成所有步,或停止工作,或从头开始继续工作 ＊ )
      IF zdW THEN
        oJS：= 0;(＊ 如果设定自动工作,则从头开始继续工作＊ )
      ELSE
        wO：= FALSE;(＊ 如果不是设定自动工作,则停止工作＊ )
```

```
            END_IF;
        END_IF;
    END_IF;
ELSE
    oJS:=0;(* 如果工作停止,计数器复位* )
    oW:=0;(* 如果工作停止,停止虚拟输出* )
END_IF;
```

（3）功能块调用如图 3-95 所示。其地址使用与图 3-93 相同。

3. 分支混合控制逻辑

混合控制也可有分支，而且也还可有不同的分支结构。图 3-96 所示为较常用的一种分支算法框图。

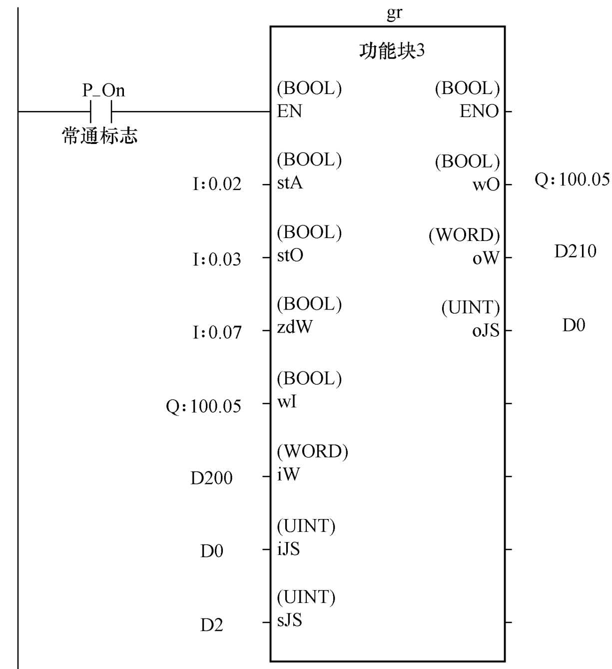


图 3-95　功能块 3 调用程序

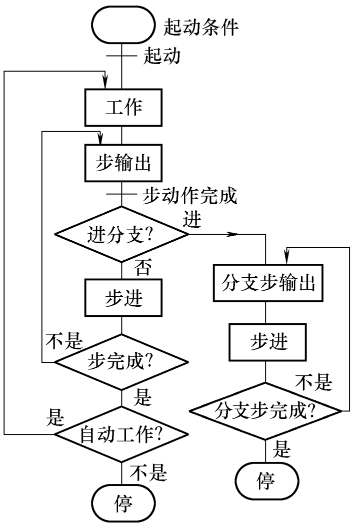


图 3-96　分支算法的框图

从图知，它在主干程序中，每次步动作完成，都要判是否进入分支程序。如进入分支程序，一旦分支步完成，则工作停止。而若未进入分支程序，如自动工作，将周而复始地执行主干程序。一般讲，这里的主干程序用于系统正常工作，而分支程序则当系统不正常时才使用。

可在图 3-96 基础上，增加反馈输入通道（字），判断是否进入分支。如进入分支，再对输出通道（字）的起始地址作新的设定。具体程序略。

4. 混合控制应用实例

（1）采用的是本章 3. 3. 4 节 3. 组合机床动力头运动控制，其要求与其完全一样。

本例用图 3-93 程序。并新增图 3-97 所示的实际输入输出程序。图中“虚拟输出”实际地址设为 220。“虚拟输入”实际地址设为 200。实际输入、输出用符号地址。用的是本章

3.3.4 节 3. 组合机床动力头运动控制，其要求与其完全一样。

图 3-97 为新增的实际输入输出程序。

本例有 6 个工作步，故 DM997 设为 #5，从 0 开始到 5，正好六步。

DM990、DM991 及 DM992 可任意设，只要所设的数据不被覆盖即可。本例 DM990、DM991 及 DM992 分别设为 #300、#200 及 #400。即虚拟输出设定值地址，从 DM0300 开始；虚拟输入有效位设定值地址，从 DM0200 开始；虚拟输入 ON 还是 OFF 有效设定值地址，从 DM0400 开始。

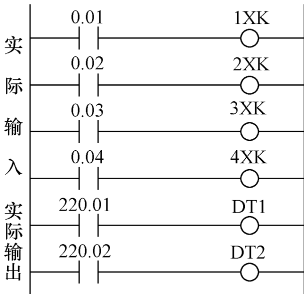


图 3-97 输入输出程序

DM0300 ~ DM0305 依各步要求的虚拟输出设定。

DM0200 ~ DM0205 依各步的有效输入位设定。

DM0400 ~ DM0405 依各步要求的 ON、OFF 有效设定。

这些 DM 区具体设定值，见表 3-23。

表 3-23 参数选择

变量地址	变量值	变量地址	变量值
DM0200	#4	DM0400	#0
DM0201	#8	DM0401	#0
DM0202	#4	DM0402	#1
DM0203	#8	DM0403	#0
DM0204	#10	DM0404	#0
DM0205	#2	DM0405	#0
DM0300	#6	DM0990	#300
DM0301	#2	DM0991	#200
DM0302	#4	DM0992	#400
DM0303	#6		
DM0304	#2		
DM0305	#4	DM0907	#5

作了以上设定后，再运行图 3-93 加图 3-97 梯形图程序，经测试证明，它完全可实现所要求的功能。

（2）设计要求同本章 3.3.4 节 1. 液体混合罐工作控制。

这里的关键是，把工作过程分成“步”，每“步”对应一个输出，然后用输入对“步”作控制。本例共分 5 步：

打开阀门 X1(10.01)ON，直到行程开关 I(0.01)ON；

关闭 X1，同时打开阀门 X2(10.02)ON，直到行程开关 H(0.02)ON；

关闭 X2，接通 M(10.03)，使 M 工作，保持 6s；

M 工作 6s（用定时器 TIM 10）后，停止工作，并打开阀门 X3(10.04)，到行程开关 L(0.01）从 ON 到 OFF。

延时 2s（用定时器 TIM 11），工作停止。

本例也是用图 3-93 梯形图程序。但也要增加实际输入、输出及定时逻辑。如图 3-98 所示。

此外，还要对有关 DM 区作设定。具体是：

DM997 因工作步数为 5，故设其为 4。

DM990、DM991 及 DM992 也可任意设。本例 DM990、DM991 及 DM992 分别设为 #900、#910 及 #9200。即虚拟输出设定值地址，从 DM0900 开始；虚拟输入有效位设定值地址，从 DM0910 开始；虚拟输入 ON 还是 OFF 有效设定值地址，从 DM0920 开始。

DM0900 到 DM0904 依各步要求的虚拟输出设定。

DM0910 到 DM0914 依各步的有效输入位设定。

DM0920 到 DM0924 依各步要求的 ON、OFF 有效设定。

这些 DM 区具体设定值，见表 3-24。

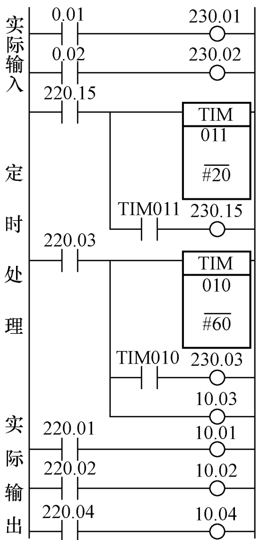


图 3-98　液体混合罐工作控制新梯形图程序

表 3-24　参数选定

变量地址	变量值	变量地址	变量值
DM0900	#2	DM0920	#0
DM0901	#4	DM0921	#0
DM0902	#8	DM0922	#0
DM0903	#10	DM0923	#1
DM0904	#8000	DM0924	#0
DM0910	#2	DM0990	#900
DM0911	#4	DM0991	#910
DM0912	#8	DM0992	#920
DM0913	#2		
DM0914	#8000	DM0907	#4

增加了以上实际输入、输出及定时程序，再按表 3-24 对 DM 区作了设定，运行图图 3-93、3-98 程序梯形图程序，完全可实现设计的要求。实际测试结果也证明了这一点。

应指出的是，这里的步进逻辑不一定非用计数器不可。也可用加 1 指令或普通的加指令。只是，用它时，在每进行一次步进，还要判断步进是否完成，没有用计数器方便。

混合控制算法及其程序不仅规范，而且“万能”。避开了麻烦的时序逻辑处理，可按部就班地实施顺序控制。动作及反馈的不同只是输出输入变换的不同。而动作增减，只须改变计数器的设定值。它所控制的点数、步数几乎不受限制。惟一的限制是 PLC 的数据区的容量及 PLC 的实际输入输出点数。

结语

本章介绍了顺序控制的概念、类型及程序分析与设计方法。顺序控制是 PLC 的优势，

也是 PLC 最基本的应用。所以，顺序控制编程是 PLC 最常用的编程。

本章介绍的顺序控制编程方法有 4 种：基本逻辑设计法、图解设计法、高级逻辑设计法及工程设计法。

基本逻辑设计最简单的为组合逻辑综合。其次为异步时序逻辑综合及同步逻辑综合。逻辑设计法比较严谨，但不易掌握。

图解设计法比较直观、好掌握，但不是所有的顺序控制都可用它编程。

高级逻辑设计法有：标志值法、多位逻辑设计法、流程控制及功能块设计法。高级逻辑设计法可充分利用 PLC 的资源，程序效率高。

工程设计法使用的指令及软器件会多些，但比较简明。可根据问题的性质，选用分散控制、集中控制或混合控制算法。这 3 个算法以混合控制算法功能最强。缩编的程序不仅效率高，而且几乎是万能的。

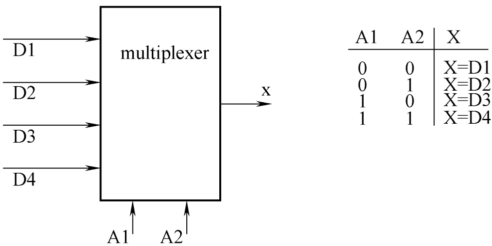
在本章介绍的算法实例化中，用了间接地址访问，还用了一些较复杂的指令及功能块。所以，在本章的学习中，还可加深对第 2 章的内容的理解。

请想想

1. 逻辑量、逻辑量控制含义？它与顺序控制的关系？
2. 比较组合逻辑与时序逻辑的特点。
3. 比较异步逻辑与同步时序逻辑的异同。
4. 比较问题的分析与综合的差别。
5. 真值表、通电表、状态图的含义与作用？
6. 组合逻辑综合要点？
7. 异步时序逻辑综合要点？
8. 同步时序逻辑综合要点？
9. 标志逻辑、多位逻辑的要点、特点及其应用？
10. 多位逻辑是什么？怎么使用？
11. 逻辑量控制工程设计方法类型及其要点？

请试试

1. 用组合逻辑设计实现多路转换器的 PLC 程序。要求如图所示 3-99。



A1	A2	X
0	0	X=D1
0	1	X=D2
1	0	X=D3
1	1	X=D4

图 3-99

2. 用状态图设计单按钮起停程序。
3. 设计 3 供水电动机工作自动切换程序。
4. 设计 4 人抢答器程序。
5. 设计用一个按钮，依次按一定规律长（2s 以上）按、短（1s 以内）按后，开锁程序。
6. 其它实际控制程序。

第 4 章　PLC 过程控制编程

过程控制主要指，对一系列模拟量所实施的控制。模拟量为连续变化的物理量，如电压、电流、温度、压力、速度、流量等。在现实世界中，特别是在连续型的生产，如化工生产中，常见到模拟量，并要求对其进行控制。

当然，过程控制也要用到开关量，其控制方法见第 3 章介绍。此外，由于脉冲技术的进步，在过程控制中，有时也用到脉冲量。但脉冲量用的最多的还是运动控制，故有关它的内容将在介绍运动控制时一并说明。

过程控制是基于信息采集和处理、使系统的状态与行为产生所希望的变化，而施加给系统的作用。这里的信息有 3 种：

一为被调节量，或称被控量，也称调节量，是反映的被控系统的状态、行为、性能或功能的信息；

另一为控制量，也称控制，即施加给系统的作用，是经 PLC 处理后产生的控制信息。

此外，还有干扰量。它与控制相反，是使系统的状态与行为产生所不希望的变化。干扰信息多不大好检测，而如采用闭环控制也可不检测。

模拟量控制实质就是从数量上对这三个信息进行变换与处理，以确保调节量能按期望的要求变化。

4.1　过程控制概述

关键词：模拟量、过程控制、调节量、控制量、扰动量、开环、闭环、单回路、多回路、自动调节、随动控制、程序控制、最优控制系统、自适应控制、时域指标、频域时域、频域指标、串级控制、前馈控制、比值控制

4.1.1　PLC 模拟量控制过程

模拟量是连续量，而且多数是非电量。而 PLC 只能处理数字量、电量。为进行模拟量控制，一般讲：

要有传感器，以把模拟量转换成电量。如果这电量不是标准的，还需要有变送器、以把电量变换为标准的电信号，如 4 ~ 20mA、1 ~ 5V、0 ~ 10V 等等。

要有模拟量（A）到数字量（D）转换的模拟量输入单元（内插板或模块），以把这些标准的电信号变换成数字信号。

要有数字量（D）到模拟量（A）转换的模拟量输出单元（内插板或模块），以把 PLC 处理后的数字量变换成模拟量。

要有执行器，根据模拟量的大小执行相应的模拟量输出控制动作。

当然，如同处理开关量一样，PLC 的 CPU、内存、相应的程序等也是必需的。只是这里多了以上提到的信号的采集、转换、变换及执行等环节。

所以，一个完整的模拟量 PLC 控制，一般来讲，其过程是：

用传感器采集信息，并把它变换成标准电信号，进而送给模拟量输入单元；

模拟量输入单元把标准电信号转换成 CPU 可处理的数字信息；

CPU 按要求对此信息进行处理，产生相应的控制信息，并传送给模拟量输出单元；

模拟量输出单元得到控制信息后，经变换，再以标准信号的形式传给执行器；

执行器的驱动系统对此信号进行放大和变换，进而使执行器产生控制作用，施加到受控对象上。

图 4-1 示出了以上介绍的模拟量控制过程。

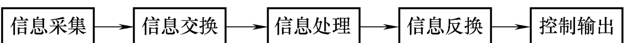


图 4-1 模拟量控制过程

需要弄清的是，这里“基于信息采集和处理”的信息，是调节量，还是干扰量。

如用的信息为调节量，则要用反馈控制。它是一种模拟量最基本的控制方式。它依据系统的实际输出与预期输出间的偏差来进行控制，以期逐步缩小这一偏差。至于产生偏差的原因，它是不理睬的。图 4-2 所示的就是反馈控制的原理图。

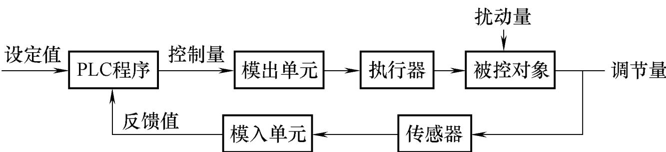


图 4-2 模拟量反馈控制

如图示，这里的传感器不断地监测被控对象的调节量，接着把它送给模拟量输入单元。PLC 程序把模拟量输入单元送来的反馈值与系统预定的设定值进行比较，进而产生控制量。这控制量再经模拟量输出单元、执行器作用到被控对象上，其目的是尽快地缩小这个差值。可知，这里的信息流是闭合的，所以反馈控制又称闭环控制。

如用的信息为干扰量，则可用前馈控制。它基于扰动补偿原理，根据扰动的情况作相应控制。图 4-3 所示为它的工作原理图。



图 4-3 模拟量前馈控制

从图中可知，它的传感器监测的是扰动量。PLC 程序根据扰动量、控制量与调节量间的关系产生相应的控制量，进而再通过模拟量输出单元、执行器作用到被控对象上，其目的是在干扰量作用于系统的同时，这个控制量也作用于该系统，以补偿干扰对系统的不利影响。可知，这里的信息流是开路的，所以前馈控制又称开环控制。

开环控制使系统在偏差即将发生之前就注意纠正偏差，这是它的优点。但要弄清有多少扰动量，以及它与调节量间的关系，即控制量随扰动变化的规律，是不易的。这也是它用得不多的原因。

以上讨论的是完整的模拟量控制过程，是较复杂的。既有模入（AI），又有模拟出

（AO）。有时，为了简单，可不用那么完整的模拟量控制。如有的只用模入，而输出用逻辑量（DO）。如控制电炉温度，简单的办法是，不停地读入温度值，并与设定值比较。如实际温度小于设定值，则控制一个逻辑量 ON，用其使加热器得电。反之，如实际温度大于设定值，则控制这个逻辑量 OFF，用其使加热器失电。再如，也可能不用模入，而用逻辑量入（DI），但用模拟量输出。再就是，由于脉冲技术的发展，模拟量控制也可运用有关脉冲控制技术。

提示：新出现回路控制模块或过程控制 CPU（即 PPC），可代替 PLC 的 CPU 实现几十路，甚至更多回路的模拟量信息处理，使 PLC 处理模拟量的能力又有了新发展。

4.1.2　PLC 过程控制目的

（1）使系统的某个量保持恒值，即要求可控系统在受到扰动时，其调节量仍能保持在设定值附近，基本上不变。这种控制称为镇定控制，或称自动调节。

（2）使系统的状态按预先给定的方式随时间，或按预定的程序变化，这种控制称为程序控制。

（3）使系统的状态按外来信号的变化而变化，这种控制称为随动控制。随动控制在实施控制以前不知道控制程序所需要的全部信息，但可以在控制系统运行期间获得这些必要的信息。

（4）使控制系统在满足一组约束条件下，目标函数值取极大值或极小值，即使系统的某一参数达到最优值，这种控制称为最优控制。

（5）使系统适应内外环境的变化，始终处于最有利的状态下运行，这种控制称为自适应控制。自适应控制往往需要一个学习和记忆的过程，通常采用搜索法来选择系统最有利的运行状态。

（6）使系统在对抗中取胜。在军事、经济、生态等系统中存在着竞争现象。这种系统往往出现两个受控部分的交互作用，在实施控制时要考虑对方的反作用，因而控制策略由两部分组成：一是要对竞争中出现的情况迅速做出反应。二是采用最优策略使系统在对方施加最不利的影响时也能处于尽可能好的地位。

虽然以上介绍了模拟量控制的六种目的，但最基本的、最常用的是自动调节。在自动调节的基础上，如设定值是随时间按要求变化的，则变为程序控制系统；如设定值是本系统外的物理量随机确定的，则变为随动控制系统；如这些系统的控制规律、或控制参数可变的，并追求在满足一组约束条件下，目标函数值取极大值或极小值，则可能变为最优控制系统，或自适应控制。

由于 PLC 是基于计算机技术的控制器，有很强的数字处理与逻辑处理功能，所以，只要有合适的算法，以上讲的多数控制都是可以实现的。

只是算法设计得有相应的自控知识，所以，模拟量控制编程，与其说是取决于设计者对 PLC 的了解，不如说是取决于设计者对自控知识的掌握；它的难点，似乎不在于 PLC 程序本身，而在于要很好地运用好有关自控知识。

4.1.3　PLC 过程控制类型

可以从不同角度划分 PLC 对模拟量控制的类型。

（1）按控制方法分，有：单回路反馈控制，串级控制等。

（2）按控制方式分，有开环控制系统，闭环控制系统及复合控制系统。

（3）按元件类型分，有机械系统、电气系统、机电系统、液压系统及气动系统。

（4）按系统功能分，有温度控制系统、压力控制系统、流量控制系统、位置控制系统及速度控制系统。

（5）按系统性能分，有线性系统和非线性系统，定常系统和时变系统。

（6）按控制要求分，有定值控制系统（调节器），随动系统及程序控制系统。

（7）按输入方法分，有模拟量入单元输入（AI）；用脉冲信号输入。

（8）按输出方法分，有模拟量输出单元输出（AO）；开关量的 ON/OFF 输出（DO）；脉冲量（不同脉宽）输出（PO）。

把以上情况进行组合，将有：AI-AO、AI-DO、AI-PO、PI-AO、PI-DO、PI-PO。此外，还可能单纯地用开关量输入（DI）控制模拟量输出（AO）。这一般是用于手动控制。

以下对各类模拟量控制作简要讨论。

1. 单回路反馈控制

它只有一个控制回路，是闭环的。具体有：

ON/OFF 控制，最简单。其办法是，把检测到的模拟量实际值与设定进行比较，当实际值超过时定值到某界限时，其执行回路 ON（或 OFF）；而低过某界限时，执行回路 OFF（或 ON）。也可用偏差信号，处理后（也可为 PID 处理，见后）的偏差信号调节控制输出 ON 与 OFF 的比例，以实现控制。这种控制仅输入用模拟单元，而输出则用开关量。

P（比例）I（积分）D（微分）控制，它由传感器、模入单元、PLC 程序、模出单元（或逻辑量输出点）及执行器组成。它对偏差作 PID 运算，然后产生控制输出。当然，也可只有 P，或 PI 的控制。视系统的要求而定。

PID 运算可用 PLC 的数学运算指令实现。较高档的 PLC 多有 PID 指令，或 PID 函数块，则可直接用这个指令，或调用这个函数块实现。也可使用 PID 控制的硬件单元（模块）实现。

其它控制，如模糊控制，它的输出按其与输入对应的模糊关系确定。OMRON 就有模糊控制单元，可用以实现这种控制。

单回路反馈控制简单，易于调整、投运，适用于纯滞后和惯性较小、负荷和干扰变化比较平缓的系统的控制。

2. 串级控制

它有主副两个控制回路，主回路与副回路，如图 4-4 所示。从图知，它的主回路的设定值按要求给定，其输出不用以推动执行器，而用作副调节器设定值。副调节器的输出才用以

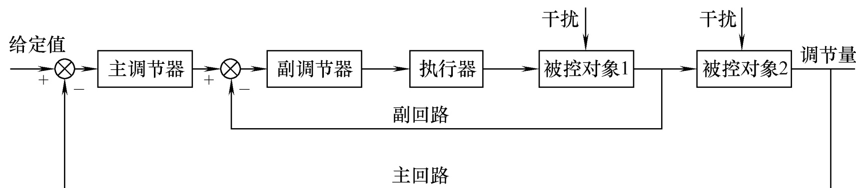


图 4-4　串级控制

推动执行器。

图 4-5 说明单回路控制与串级控制的区别。图 4-5a 为单回路控制，控制量为炉温 θ，测出的炉温经变送器变换后，送调节器 T，调节器的输出直接控制调节阀的开度，从而控制送入加热炉中的燃油量。这个系统虽较简单，但燃油压力的波动将影响炉温变化。特别是燃油的压力波动值大、且频繁时，炉温的波动更大。

而图 4-5b 为串级控制，就是要解决这个压力波动对炉温的干扰。它先构成一个压力控制回路。用副调节器 G，去克服燃油压力波动对流量的影响。主调节器则按设定值的要求，依炉温，去确定应给的燃油流量。

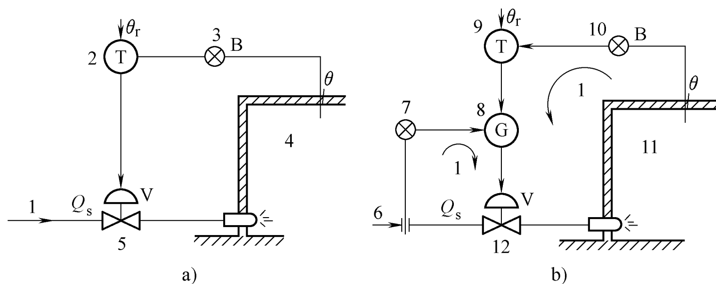


图 4-5　单回路控制与串级控制区别

a）单回路控制　b）串级控制

1、6—燃油　2—调节器　3—变送器　4、11—加热炉　5、12—调节阀

7—副变送器　8—副调节器　9—主调节器　10—主变送器

由于多了个副回路，可使所控制的炉温免受或少受燃油压力波动的干扰，从而提高系统的控制品质。

应提醒的是，这里调节器用的不是硬件，而是软件、是 PLC 程序。靠运行 PLC 程序实现调节器的控制功能。

3. 前馈控制

图 4-6 为前馈控制的例子。它是按扰动进行的开环控制。如图示的换热器，加热的物料流入量是主要的干扰因数时，可用如图所示的办法，随着进料的变化，通过前馈补偿器，调节用以加热物料的蒸汽流量，从而控制容器的温度。

如果弄清物料流量对温度的影响规律，可作到系统的误差为零。

当然，前馈与反馈控制也可结合起来进行，以得到更高系统的控制品质。

应提醒的是，这里的前馈补偿器也不是硬件，而是软件、也是 PLC 程序。靠运行 PLC 程序实现前馈补偿器的控制功能。

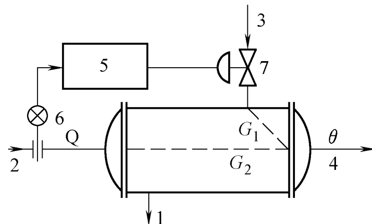


图 4-6　前馈控制例子

1—排水　2—进料　3—蒸气

4—出料　5—前馈补偿装置

6—进料变送器　7—调节阀

4. 比值控制

在生产中，有时要求若干变量间保持一定的比例关系，如煤气加热炉，就要求煤气与空气要有合适的比例，即空燃比。比例调节器就是要保证在煤气变化的同时，空气也要有相应的变化。比值控制有

开环、闭环及多变量比值等。

图 4-7 所示为开环比值控制。Q_b 的流量按比例 k，跟随流量 Q_a 变化。

图 4-8 为闭环比值控制。Q_b 的流量为闭路控制。但，它的设定值为 kQ_a，随 Q_a 而变。

图 4-9 所示为双闭环控制，两个回路都是闭环的。Q_a 调节器的设定值是独立的，而 Q_b 调节器的设定值由 Q_a 的变送器，经比例器换算后给出。这种空置的目的是 Q_b 要随 Q_a 变，以保证其间比例关系不变。

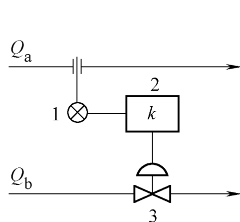


图 4-7 开环比值控制
1—变送器 2—比例器
3—调节阀

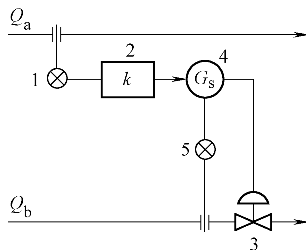


图 4-8 闭环比值控制
1、5—变送器 2—比例器
3—调节阀 4—调节器

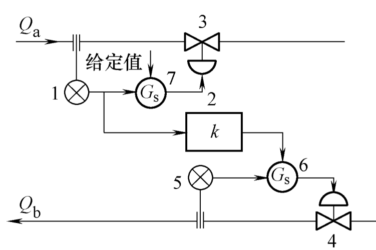


图 4-9 双闭环比值控制
1、5—变送器 2—比例器 3、4—调节阀 6、7—调节器

图 4-10 所示为多值比例控制。这里画出两个闭环控制回路。它们的调节器的设定值都是由 Q_a 的变送器送出、经比例器 k_1、k_2 换算后确定。以此保证 Q_a、Q_{b1}、Q_{b2}之间的比例关系。

图 4-11 所示也为多值比例控制。这里也画出两个闭环控制回路。Q_{b1}调节器的设定值都是由 Q_a 的变送器送出、经比例器 k_1 换算后确定。而 Q_{b2}调节器的设定值则是由 Q_{b1}的变送器送出、经比例器 k_2/k_1 换算后确定。它是用从变量间的协调，保证 Q_a 与 Q_{b1}以及 Q_{b1}与 Q_{b2}之间的比例关系。

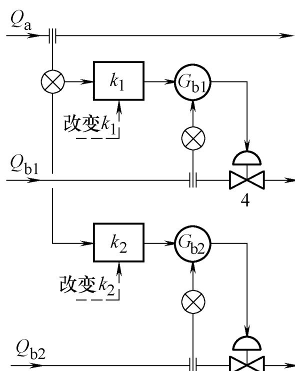


图 4-10 多值比例控制

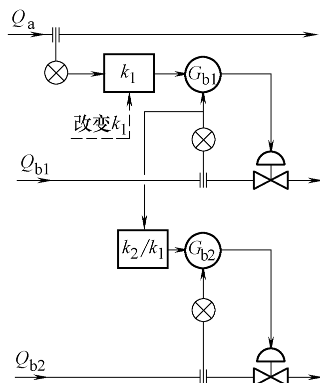


图 4-11 多值比例控制另例

图 4-12 有 3 个调节回路。调节器 T，用以调节炉温。当温度变化时，它改变燃气调节器的设定值。进而改变燃气流量 Q_a。而当燃气流量改变时，经比例器 k 也改变空气调节器的设定值。也会改变空气的流量 Q_b。从而达到当温度变化时，既改变燃气，又改变空气的流量，以保证炉温恒定。

图 4-13 变比值的控制回路。流量 Q_b 的调节器的设定值由乘法器给定。乘法器进行

k（c）与 Q_a 的乘运算。而 k（c）则由成分分析仪测出成分值 c 后，由控制器 A 把其与设定值 c_r 比较、换算得出。这种控制可保证不同的流量 Q_a 时，将有不同比例的 Q_b 值。

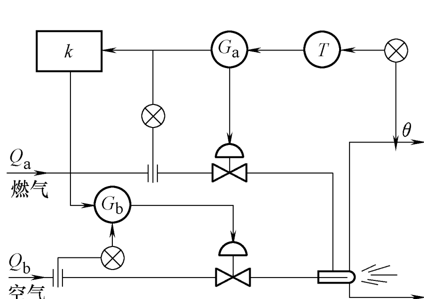


图 4-12　三调节回路比值控制

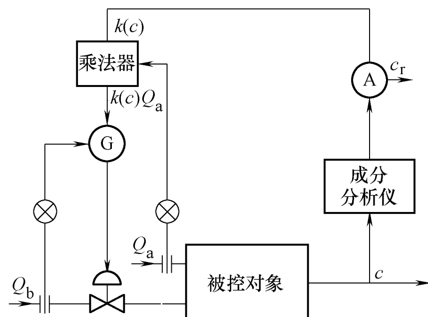


图 4-13　变比值控制

应提醒的也是，这里除变送器、调节阀之外，其它的如调节器、比利器、成分分析仪、乘法器等均为软件，为 PLC 程序，靠运行 PLC 的这个程序实现有关的控制功能。

5. 其它控制

其它常用的控制方法还有均匀控制、分程控制、多冲量控制等。

均匀控制用于连续生产的过程中。目的是保证，前后设备间的物料流动能得以平衡，以达到均匀生产的目的。

分程控制用于有不同工况的生产过程。可作到在各个工况下，都能实现合适的控制。

多冲量控制用于有多个相互有联系的被控对象，被控量不仅与控制量有关，还与其它变量有关。多冲量控制就是将这些变量组合起来，一起去控制控制量等等。

此外，还有一些高级控制，如模糊控制、专家控制、最优控制、自适应控制、自学习控制、预测控制及复合控制等等。

这些控制也都可用 PLC 予以实现。

4.1.4　PLC 过程控制特点

用 PLC 实现模拟量控制有 3 个基本特点：一是有误差；二是断续的；三是有时延。在解析这 3 个特点前，以下先对物理量在时间上、取值上的特点作以下说明。

1. 误差

模拟量在时间上、取值上都是连续的。

对模拟量按一定时间间隔取值，称为采样。采样后得到的量即为离散量。显然，离散量在时间上是离散的、即只在采样的瞬间代表当时的模拟量值，其它时刻的模拟量值不代表。但取值上是连续的。

用数值来逼近离散量，即求出与实际的离散量最接近的数字量，称为量化。量化后的离散量称为数字量。数字量在时间上与取值上都是离散的。

PLC 只能处理数字量，用它控制模拟量，必须先对这些模拟量进行采样与量化。

正是要量化，所以量化后的值与模拟量的原值总是有差异的。即存在误差。但这个误差是可控的。办法靠选用合适的模入、模出模块的位数。如用的是 8 位模入模块，其量化的值只能是 0～255（十六进制 FF）之间的整数。故其分辨率为 1/256。如用的是 12 位模入模

块，其量化的值只能是 0 ~4095（十六进制 FFF）之间的整数。故其分辨率为 1/4096。如果选的位数多，分辨率高，精度也高。但位数多，模块也贵。高过 16 位时，还要用双字指令处理，但这也将多增加资源开销与处理时间。

误差可得到控制是一个重要的优点。历史上出现用数字计算机代替模拟计算机，正是前者的误差是可控的。靠模控制的金属切削机床被数控机床所代替，原因之一也与着有关。所以，这里量化后有误差可能还是它的优点。

只是这里也有一个合理的“度”，应在保证精度的要求下，力争减少位数。

2. 断续

正是要采样，所以它是断续的。只是在 PLC I/O 刷新时，模入模块才把实际值读入 PLC；模出模块或输出点才把控制信号输出给控制对象。只是在这时，才相当于它的采样开关合上，系统是闭合的。但这个闭合时间是很短暂的。而较长的时间是 PLC 运行程序及处理采集到的数据。而这期间系统闭环是断开的。可知，PLC 模拟量控制系统使典型的采样控制系统。

为了保证采样信号能较少失真地恢复为原来的连续信号，根据采样定理，采样频率一般应大或等于系统最大频率的两倍。而最大频率是指系统幅频特性上幅值为零时的频率。由于 PLC 工作速度很快，一般讲，这个要求总是能够满足的。

3. 时延

实际系统本身的惯性以及动作传递也有个过程，有一定时延。用 PLC 进行控制，采样、信息处理及控制输出也有个过程，更有时延。在实施一个新一轮的控制作用之后，不能指望立即就会有所反应。所以，不能因一时未得到所期望的反应，就一味地改变控制作用。那样，很可能使系统出现不稳定。再如，用 PID 控制，其运算间隔时间不能太短。如无特殊措施，其间隔起码要大于程序的扫描周期等。

以上 3 个特点，在确定控制算法、设计控制程序及选定控制参数时，必须考虑到的。

4.1.5 PLC 过程控制要求

一般讲，对 PLC 模拟量控制系统的要求，都是看在某种典型输入信号下，其被控量变化的过程。例如，自动调节系统，就是看扰动作用引起被控量变化的过程；随动系统就是看被控量，如何克服扰动影响，跟随给定量的变化而变化的过程等。对每类系统被控量变化过程的共同要求是稳定性、准确性和快速性。即：稳、准、快。

1. 稳定性

稳定性一般指，系统的被控量一旦偏离期望值，则应随时间的增长逐渐减小或趋于零。对于稳定的自动调节系统，其被控量因扰动而偏离期望值后，经过一个过渡时间，应恢复到原来的期望值；对于稳定的随动系统，被控量应能跟踪给定量的变化而变化。反之，不稳定的控制系统，其被控量偏离期望值后，将随时间的增长而发散。

所以，稳定性是保证控制系统正常工作的先决条件。稳定是所有控制系统首先要满足的要求。不稳定，被调节量老是变化不定，以至于产生振荡，那是绝对不允许的。

2. 准确性

准确性是指，当过渡过程结束后，被控量的稳态值与期望（给定）值一致性。实际上，由于系统结构，外作用形式以及摩擦、间隙等非线性因素的影响，以及受模拟量控制与数字

量相互转换分辨率的限制，被控量的稳态值与期望值之间总会有误差，称为稳态误差。在实际上，完全没有这个误差是不可能的。

这个稳态误差小，则精度高。使这个误差应尽可能小，这也对控制的基本的要求。精度当然越高越好。但也要有个合适的“度”。一般讲，合乎要求也就可以了。

3. 快速性

除了稳定性、准确性，在多数情况下，还对过渡过程的形式和快慢要有要求，一般称之为动态性能。例如，对用于稳定的高射炮射角随动系统，虽然炮身最终能跟踪目标，但如果目标变动迅速，而炮身跟踪目标所需过渡过程时间过长，就不可能击中目标；对用于稳定的自动驾驶仪系统，当飞机受阵风扰动而偏离预定航线时，具有自动使飞机恢复预定航线的能力，但在恢复过程中，如果机身摇晃幅度过大，或恢复速度过快，就会使乘员感到不适；函数记录仪记录输入电压时，如果记录笔移动很慢或摆动幅度过大，不仅使记录曲线失真，而且还会损坏记录笔，或使电器元件承受过电压等等。

总之，快速性是指系统实际值偏离要求（设定）值时，系统能很快（过渡时间短）而又平稳（无振荡，或振荡幅度小、次数少）地回到设定值。

控制过程品质的定量指标有：时域指标、频域指标及积分指标。

4. 时域指标

它用在设定值阶跃扰动时，系统的控制量随时间的变化过程反映控制品质。图 4-14 所示的即为这个过程。从图可看出控制量在时域上的变化过程。

图中表示的有关时域品质指标有：

（1）最大动态偏差，图中为 B_1，及百分率超调量 σ。后者用最大动态偏差与稳态值之比：

$$\sigma = (B_1 \div D) \times 100\%$$

这个偏差当然越小越好。

（2）调整时间 t_s。从干扰发生起到控制量与其新的稳定值之差在 ±(2～5)% 之间，并不再超出这个范围所经历的时间，图中为 t_s。这个时间当然越短越好。

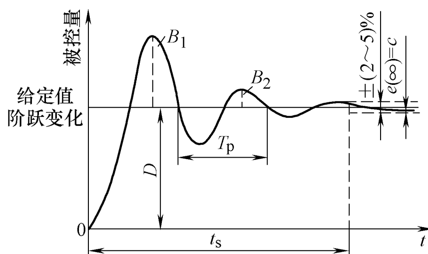


图 4-14　控制量在时域上的变化过程

（3）静态偏差 e。过渡过程结束后，被控量与设定值之差，图中为 e。这个偏差当然越小越好。

（4）衰减率 ψ。过渡过程出现震荡时，过渡过程中出现的第一与第二个峰值之比，即：

$$\psi = 1 - (B_2 \div B_1)$$

这个衰减率当然越大越好。

（5）震荡周期 T_p。过渡过程出现震荡时的震荡周期，图中为 T_p。这个周期当然越短越好。

时域指标较直观，比较好理解，但难于直接求出它的具体的值。所以，在多数情况下，还是要用频域指标。

5. 频域指标

它用系统的开环频率特性代表控制品质。具体是，在不同频率的单位信号作用下，系统开环时，其输出对信号的响应特性，即系统输出的幅值及它与作用信号的相位差。图 4-15

为系统开环频率特性图，图 4-16 为幅频特性曲线。从图可看出控制量在频域上的变化过程。

用它代表控制品质的指标有：

（1）增益裕度和相角裕度，这两个是衡量控制系统稳定度的指标。如图 4-15 所示，增益裕度为矢量轨迹与实轴相交的那一点的增益的倒数，为 $1/\alpha$。相角裕度为矢量轨迹上增益为一的点偏离负实轴的角度，图中为 γ 角。

（2）共振频率。相应于系统幅频特性上增益最大时的频率。图 4-16 其上的 ω_r 即为共振频率。

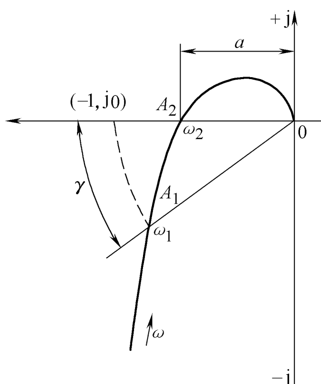


图 4-15　增益裕度和相角裕度

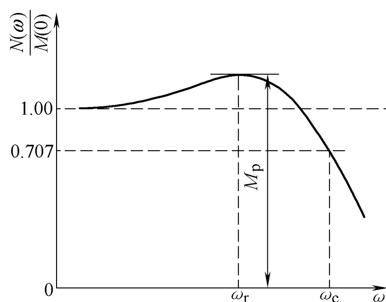


图 4-16　幅频特性

（3）截止频率。增益比为 0.707 时的频率，图中为 ω_c。

（4）最大增益比。幅频特性中增益最大的值，图中为 M_p。

6. 积分指标

用积分指标可综合反映控制系统的稳定性、准确性及快速性。特别是用计算机分析系统时更为有效。常用的指标有：

（1）误差 $e(t)$ 积分值。

（2）绝对误差 $|e(t)|$ 积分值。

以上两项综合反映了过渡过程的时间与偏差。

（3）误差（$e(t)$）平方积分值。

（4）绝对误差 $|e(t)|$ 和时间乘之积分值。

为使控制系统具有较高的稳定性、准确性及快速性，所有这些值都应尽可能的小。一般控制有这个要求，用 PLC 控制也不例外。

4.2　PLC 模拟量输入及输出

关键词：模拟量传感器、模拟量输入、模拟量输出、模拟量执行器

4.2.1　模拟量传感器

传感器（transducer）是按一定规律实现信号检测并将被测量（物理的、化学的和生物的信息）变换为另一种物理量（通常是电量）的器件或仪表。它既能把非电量变换为电量，

也能实现电量之间或非电量与电量之间的互相转换。

在生产过程控制系统中，常把输出为标准信号的传感器称为变送器，而把输出为非标准信号的传感器称为敏感器。

传感器是 PLC 对模拟量控制要用到的关键部件。没有它，无法检测到信息，基于信息也就不可能实现，也无从达到调节、控制的目的。

1. 传感器类型

传感器品种繁多，原理各异。它们是根据不同需要和不同测量对象研制出来的。传感器的分类方法很多。

根据现象所属领域不同可分为物理传感器、化学传感器和生物传感器。

根据所用敏感元件的材料来分，有半导体传感器、陶瓷传感器、有机高分子传感器、光纤传感器等。

根据功能来分，有单功能传感器、多功能传感器、智能传感器、仿生传感器等。

根据构成原理可分为结构型传感器、物性型传感器和智能型传感器等。

（1）结构型传感器。利用物理学中场的定律（包括电场、磁场、力场等）构成的传感器。它的基本原理是以部分结构的位置变化和场的变化来反映被测非电量的大小及其变化。结构型传感器大都采用机电结构和间接信号变换方式。所谓间接变换就是信号经过两次变换，先将被测信号经过机械式检出元件转换成中间信号，然后再经过敏感元件转换成电信号输出。例如应变电阻式压力传感器就是通过弹性膜片把被测压力检测出来并变换为应变值，再用应变电阻元件把应变值变换为便于处理的电信号输出。

结构型传感器应用最广，采用的测量原理主要有电磁检测、光电检测等。

1）电磁检测：包括电阻式传感器、电感式传感器、电容式传感器、电涡流式传感器压电式传感器、热电式传感器、压阻式传感器、压磁式传感器、霍尔式传感器（见霍尔式压力传感器）、频率式传感器（见谐振式传感器）、数字式传感器（见感应同步器、磁栅式传感器）。

2）光电检测：包括光电式传感器、激光传感器、红外线传感器、光栅式传感器、光纤传感器等。

3）此外还有超声波传感器、核辐射传感器以及用电化学、核磁共振等方法制成的传感器。结构型传感器中直接感受被测非电量的敏感元件是机械式元件和电磁式元件，其中弹性敏感元件应用最广。它把各种形式的非电量转换为应变量或位移量。如果弹性敏感元件的输出是应变，则可能制成各种形式的应变传感器，如果弹性敏感元件的输出量是位移（线位移或角位移），则可能制成电感式、电容式、电涡流式或电阻式传感器。

（2）物性型传感器。利用物质特性（包括各种物理、化学、生物的效应和现象）构成的传感器。它的基本特征与构成传感器敏感材料的特性密切相关。物性型传感器采用直接信号变换方式，就是用一种敏感元件将被测信号直接转变为电信号输出。这是发展最快和引人注目的新型传感器。利用物质的化学特性构成的传感器称为化学传感器，利用生物学特性的传感器称为生物传感器。这两种传感器都属于物性型传感器，它们研制困难但性能优越，发展潜力很大。

固体敏感元件是物性型传感器的关键组成部分。对各种具有敏感功能的材料的研究，即如何利用它们的物理、化学、生物学的特性和效应来构成固体敏感元件，已经形成一门新技

术。根据不同应用目的，固体敏感元件通常按照检测参数和元件功能来分类。

（3）智能型传感器。它是物性传感器进一步发展的产物。智能型是指除具有检测功能外，还具有自补偿、自校正、自调整、自诊断和逻辑操作、程序控制、自动实现计量和检测最优化等功能。这些功能往往是多功能敏感元件与微处理机或单片微型计算机相结合的结果。美国霍尼韦尔公司研制的 DSTJ-3000 型差压传送器是典型的智能型传感器。它采用集成工艺技术在一块硅片上将测量差压、压力、温度的多功能敏感元件与 CMOS 微处理机结合起来。它的敏感元件是硅膜片。微处理机能在不同的温度、压力条件下作差压的补偿运算。在产品出厂时，对每台传感器进行编码，并将编码存放到存储器中，以便微处理机进行运算。精度可达 0.1%，量程比达 400:1。输出经数模转换器可变为 4 ~ 20mA 的模拟信号。它还能把数字信号叠加到模拟传输信号中。由于专用程序能与传感器、电源和负载阻抗任意连接，它还能双向通信并能在信号传输线的任意位置上，远距离地校准零档、调整阻尼、变更测量范围、选择输出（线性的或平方根的）和读出传感器本身的自诊断结果。

2. 性能指标

传感器的性能应该根据测量目的、使用环境、被测对象、精度要求、信号处理以及成本限制等条件来确定。传感器的基本性能要求是：输出信号与输入信号成比例；迟滞和非线性误差小；内部噪声小，不易受外界干扰影响；反应速度快；动作能量小；对被测状态的影响小；使用寿命长；使用、维修和校准方便。为了比较和评价传感器的性能，人们规定出一些能给以定量描述的性能指标，常用的性能指标有 9 项。

（1）量程。测量上限与下限的代数差。

（2）测量范围。测量上限与下限之间的区间。

（3）过载。传感器在不致引起规定性能指标永久改变的条件下允许超过测量范围的能力。

（4）灵敏度。传感器输出的变化值与相应的被测量的变化值之比。

（5）分辨率。传感器可能检出的被测信号的最小增量。

（6）误差。被测量指示值与真值之间的差，传感器的基本误差一般包括重复性、非线性、迟滞等。

（7）重复性。在同一工作条件下，对被测量的同一数值在同一方向上进行重复测量时的测量结果的一致性。

（8）非线性。在规定的环境条件下，传感器校准曲线与传感器拟合直线（理想直线）的不一致程度。

（9）迟滞。当输入作全测量范围移动时，同一测量点正反行程输出的不一致性。

3. 发展趋势

传感器的发展趋势表现在采用有关学科的新技术、采用新材料、组合化和单片智能化 4 个方面。

（1）在研究各种物理化学效应的应用技术以及信号处理技术的基础上研制新型传感器，激光、超声、微波和仿生技术的利用，尤其受到人们的注意。

（2）在采用新材料、新工艺的基础上开发新型传感器。改变材料的组成、结构、添加物或采用各种工艺技术，利用材料形态变化如薄膜化、微小化、纤维化、气孔化、复合化、无孔化等，提高材料对电、磁、光、热、声、力、吸附、分离、输送载流子、化学、生物等

的敏感功能。

（3）研究传感器组合技术，提高传感器测量精度。对于有限的敏感元件，根据不同使用条件来运用各种测量技术和控制技术，如温度补偿技术、抗电磁干扰技术、高频响应技术、信号处理技术等，实现不同的组合，构成各类传感器。

（4）研究敏感元件的小型化、集成化、固体化、多功能化。研制新型场效应敏感元件、厚薄膜和超细微粒子敏感元件、色敏元件、光纤元件等，发展数字式传感器和智能型传感器，通过集成工艺实现检测、转换和信息处理一体化，最终实现传感器的单片智能化。

4.2.2　模拟量输入

有多种输入模拟量的方法：

1. 用模拟量输入单元输入模拟量

把模拟量转换成数字量的 PLC 工作单元称模入单元，简称 AD 单元。多数 PLC 的 AD 单元是单独的模块，但也有集成到 CPU 模块中的，如 OMRON 公司的 CQM1 CPU43、CP1H A 型机。

模拟量一般指标准电信号、电流或电压。电流为 4 ~ 20mA。电压为 0 ~ 10V，或 1 ~ 5V，或 ±10V 等。具体是什么，又是多少，可依型号情况及设定开关设定。

转换后的数字量可以为二进制 8 位、10 位、12 位、16 位，或更高。对应的分辨率分别为量程的 1/255、1/1023、1/4095 及 1/32767，或更小。分辨率高精度也高。大、中型机的模入单元，精度高，多为 12 位，小型机差点，不少为 8 位。

图 4-17 所示为模入单元的工作示意框图。

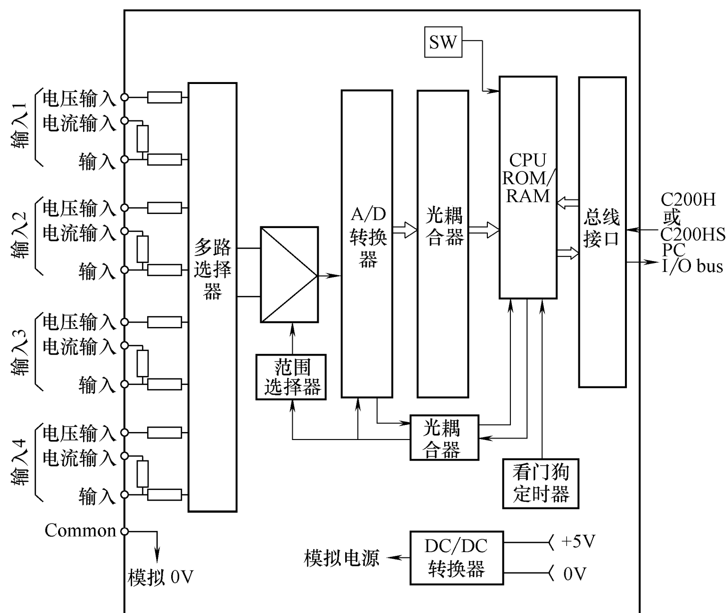


图 4-17　模入单元示意框图

从图知，它自身有输入电路（本例为四路）、多路选择器、A/D 转换器、范围选择器、光耦合器、CPU、内存、看门狗定时器、电源及总线接口。

从输入电路知，它可接电流信号，也可接电压信号。

一个模入单元一般只有一个 A/D 转换器。但有了多路选择器的依次切换，则可实现多路模拟信号处理。转换后再经光耦合器转储到它自身的内存中。这样做，当然要耽误一些时间，但节省了器件与空间。算是以时间换取空间嘛。

有的模入单元，可在存储之前进行相应的处理，处理后才存；存储后的数据，再经 PLC 的 I/O 总线接口，在 PLC I/O 刷新时，被读入到 PLC 内部继电器或 I/O 继电器的相应通道中。

由于这里用有光耦器，故与普通的 I/O 单元一样，抗干扰的能力也很强。但有的公司为了降低成本，也生产有无隔离的 AD 单元。当然，它抗干扰能力也差了。

常用的 AD 单元有 4 路、8 路的，还有多达 16 路的。也有少的只有 1 路、2 路的等等。

使用 AD 单元时，要了解它的性能。它的主要性能有：

（1）模拟量规格。指可接受或可输出的标准电流或标准电压的规格，一般多些好，便于使用；

（2）数字量位数。指数字量用多少位二进制数表达多，较好，精度高；

（3）转换路数。指可实现多少路的模拟量转换，多好，可处理多路信号；

（4）转换时间。指实现一次模拟量转换的时间，少好，路多时，时间长；

（5）功能。指除了实现数模转换时的一些附加功能，有的还有标定（Scaling）、平均（Mean）、峰值（Peak Vaule）及开方（Square Root）功能。其含义分别是：

1）标定。设定转换后的数字量的上限（与模拟量的最大值对应，如 20mA）及下限（与模拟量的最小值对应，如 4mA），当使设定标定功能使能时，则单元会自动地把模拟量按比例转换成上、下限之间的值。如图 4-18所示。

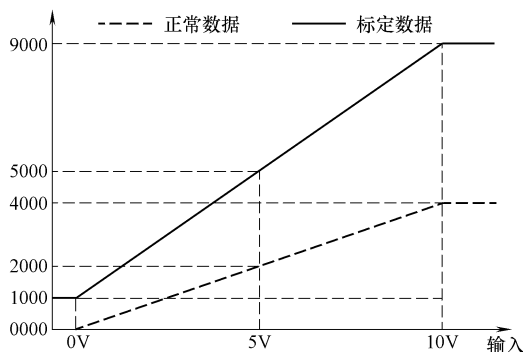


图 4-18　数据标定

图中虚线表示的是，未标定时的电压与输出值的对应关系：0V 时，输出为 0；10V 时，输出为 4000。实线为标定功能使能后的情况。这时，0V 时，输出为 1000；10V 时，输出为 9000。在 0 ~ 10V 之间，如 5V，则为 5000。当然，这个 5000，是输入单元自动给出的，人工不必计算。

2）平均：可连续采集多次数据，然后加以平均，以平均后的数作为输入。要否平均，由多少次数作平均，可设定。使用平均，可减少干扰，但转换时间将增长。

3）峰值：可保持输入过程的最大值。峰值保持使能失效，则保存值复位为零。

4）开平方：当平方根使能位使能后，可使数据转换成平方根，以便于特殊使用。

最后要说明的是，不是所有模入、模出单元都有这些功能，一般性能高的才有这些功能。虽如此，但 PLC 却都有很多相关的数据计算指令，对此，也可作处理。

使用 AD 单元第一步是选用。要选性能合适的单元，既要与 PLC 的型号相当，规格、功能也要一致，而且配套的附件或装置也要选好。

第二步是接线。要按要求接线，端子上都有标明。用电压信号，只能接电压端；用电流

信号只能接电流端。接线要注意屏蔽，以减少干扰。

第三步是设定。有硬设定及软设定。硬设定用 DIP 开关，软设定则用 DM 区，或运行相应的初始化 PLC 程序。作了设定，才能确定要使用哪些功能，选用什么样的数据转换，数据存储于什么单元等。一句话，没有进行必要的设定，如同没有接好线一样，单元也是不能使用的。

2. 用采集脉冲输入模拟量

PLC 都有接收脉冲信号的单元（中、大型机）或输入内插板或输入点（小型机）。它用的是输入中断的方法接收脉冲，计数、计数值与设定值比较以至于比较结果输出，都可用中断的方法处理。其响应速度是非常快的。所以，用它可接收与处理几 k 到几百 k 频率的脉冲信号。

如使用带有中断功能的输入点，也可编写有关中断服务程序，对输入脉冲计数及处理。其计数与处理的频率也是很高的。

如脉冲频率低于每秒 100 次，用普通的输入点，并用定时中断程序，定时查询与处理该点的输入脉冲，也是可以的。

用脉冲信号，可计脉冲数，也可计脉冲频率。前者与运动的行程、累计流量等绝对量对应，后者与运动速度、流量等相对量对应。只要模拟量传感器能把模拟量转换为脉冲量或脉冲频率，PLC 正确接收后，即可进行反变换，从测到的脉冲量或脉冲频率即可得知原始的模拟量值。

以下对定时中断程序，定时查询与处理该点的输入脉冲作简介：

OMRON CP2A 等机型有定时中断指令，其梯形图格式为



这里，C1 为控制码，取值 000（一次性定时开始），003（周期性定时开始）时，C2 为定时时间间隔，C3（BCD 码）代表将调用的子程序号；取值 006（读定时器现值）时，C2、C2 + 1 及 C3 存定时器开始定时后已过去的时间及相应单位，见以下注 3；取值 010（停止定时）时，C2 、C3 没有用，但须设为 0。

注 1：C2 可以是常数，BCD 码格式，取值可从 0000 ~ 9999，单位毫秒。但取值为 0000 时，等于不进行定时中断。C2 也可以是地址，要占两个字。低字（C2）代表定时值，BCD 码。高字（C2 + 1）为定时单位，也为 BCD 码。取值只能在 0005 ~ 0320 之间，单位为 0. 1ms；即所可能设的单位可以在 0. 5ms ~ 32ms 之间。

注 2：STIM 也是减计数计时，计时开始，STIM 现值为设定值，进而每经历一个计数单位，其现值减 1。当现值减到 0 时，将调以 C3 编号的子程序。同时，如 C1 设为 003 时，STIM 的现值又返回到设定值，并重新开始计时。

注 3：当 C1 = 006，读定时器现值时，C2 存的为定时器已过去时间数，C2 + 1 存的为时间数的单位，都是 BCD 码。C3 存的也为定时器已计过的时间数，BCD 码，但其单位为 0. 1ms。而计算从最后一次计时开始，真正已过去的时间为

$$\{(\text{C2的值}) \times (\text{C2}+1\text{的值}) + (\text{C3的值})\} \times 0.1\ \text{ms}$$

图 4-19 所示的为起动周期定时中断程序，在 PLC 第一个扫描周期时执行。其含义是，

定义一个定时值为 10ms 的，周期工作的，内部定时中断程序。每当定时到，即调子程序 1。

图 4-20 所示为子程序 1 的内容（子程序的标号 1 未标出）。这里用的是符号地址。只要这里"脉冲信号"频率，不大于定时中断时间间隔的倒数，本子程序即可把"脉冲信号"（通过"@ INC"指令），计入"即时脉冲数"字地址中，而保证不丢失脉冲。

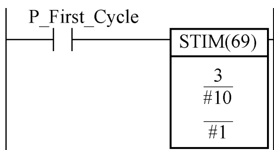


图 4-19　起动周期定时中断程序

从图还可看出，每次调本子程序一次，通过执行"INC"指令，还使"计时值"增 1。进而，执行"CMP"指令，把"计时值"与常数 10（BCD 码，当然，也可为别的什么数）比较。若"计时值"等于 10，则"P_ EQ"ON，接着使"LR0. 11"ON。"LR0. 11"ON 后，将执行随后的 3 个"MOV"指令，相继实现把"即时脉冲数"传送给"累计脉冲数"，把 0 传送给"即时脉冲数"，把 0 传送给"计时值"。后两者为新一轮累计计数作准备。

从以上讨论可知，这里"累计脉冲数"存的是每 10 个定时中断间隔计入的脉冲总数。反映的计入脉冲的频率，间接反映了如转速这类的模拟量。

应指出的是，尽管 PLC 采集脉冲信号的实时性不如采集模拟信号，但它比模拟信号电平高、信号失真的影响不大，所以，抗干扰能力特强，加上 PLC 处理脉冲技术的发展，因而，当今用脉冲量输入去反映模拟量已用的越来越多。有关处理脉冲信号更深入的问题，在本书第 5 章将作进一步讨论。

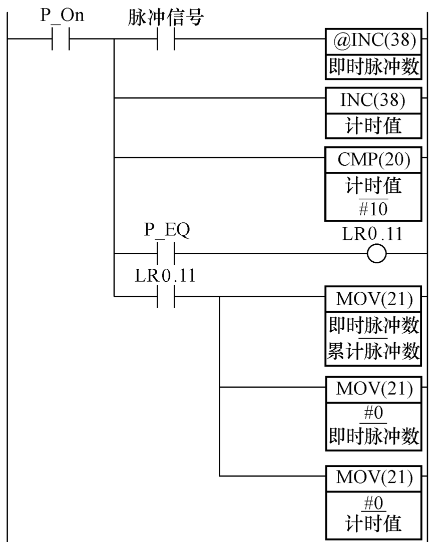


图 4-20　计脉冲频率程序

3. PLC 其它模拟量输入、输出方法

随着 PLC 模拟量控制应用的增多，除了模入、模出单元，还有模入与模出混合在一起单元，这更便于用户使用。选用一个混合模块，就可实现对若干路的模拟量控制。

此外，还有可进行温度检测、流量检测、称重检测等单元。可把检测这些物理量的传感器接入这些单元，不用变送器，即可直接实现这些物理量到数字量间的转换。有了这些模块，对这些模拟量的检测就更方便了。

再进一步，还有种种模拟量或某个物理量控制模块。不仅能检测这些物理量，而且，还可按一定算法，产生模拟量输出，不通过 PLC 的 CPU 就可实现对控制对象的控制。如 PID 控制、模糊控制模块就是这样。再如温度控制模块，实质上，它就是挂接在 PLC 上的一块温度控制表。这时，PLC 的作用只是与其交换数据与实施必要的监控。

4. 2. 3　模拟量输出

1. 用开关量 ON/OFF 比值控制输出

改变开关量 ON/OFF 比例，进而用这个开关量去控制模拟量，是模拟量控制输出的最简单的办法。如图 4-21 所示，输出的为某开关量，改变输出周期，即可调整这个输出

点 ON/OFF 的时间比例。如电源通过这个接点，加载到某模拟量控制对象，则对象所接收的能量将与这个 ON/OFF 比例相关。显然，这里改变输出周期，即控制了相关的模拟量。

图 4-22 所示为实现这个算法的梯形图程序。它用的是高速定时器。当“输出周期”小于“工作周期”时，按图 4-21 所示，部分时间有输出。当“输出周期”大于或等于“工作周期”时，全部时间都有输出。

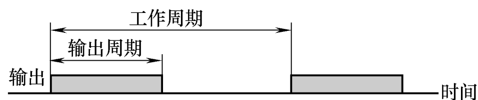


图 4-21　ON/OFF 时间比例输出

这个方法可不用模出模块，即可实现模拟量输出控制。不足的是，这个方法的控制输出是断续的，系统接收的功率有波动，不很均匀。如系统惯性较大（它对波动有滤波作用），或要求不高，容许不大的波动时，还是可用的。

为了减少波动，可缩短工作周期。但如用的 PLC，其输出点是继电器，则这个缩短是有限的。因为继电器接点通断过于频繁，将影响它的工作寿命。

2. 用可调制脉宽的脉冲量控制输出

有的 PLC 有半导体输出的输出点，则可把图 4-22 梯形图程序的工作周期缩小，以提高模拟量工作的平稳性。

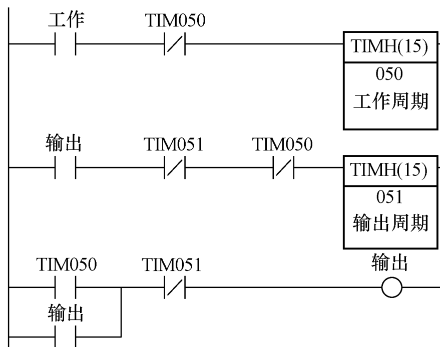
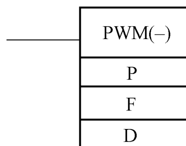


图 4-22　ON/OFF 时间比例输出程序

有的 PLC 还有产生脉冲输出的输出点，并伴有占空比（为一个脉冲循环内脉冲的 ON 时间与 OFF 时间之比）可调（脉宽调制）的脉冲输出指令。用其控制模拟量，则是既简单，而平稳性又好的方法。以 OMRON 的 CPM2A 机（但必须为半导体输出的机型，即 CPM2A-CDT-D 或 CPM2A-CDT1-D）为例，它的 010.00 及 010.01 两输出点，即可作为脉冲输出点。它还有 PWM 指令。用其去控制模拟量效果就很好。CJ1M 机也有此指令。

CPM2A 机的 PWM 指令的梯形图格式为



这里，P 为脉冲输出口地址，为 000（用口 1，输出点 010.00）或 010（用口 2，输出点 010.01），两个口可同时独立工作，互不影响；

F 为指定脉冲频率，必须为 BCD 码，在 0001 ~9999（相当于 0.1 ~999.9Hz）之间任选；

D 为占空比，必须为 BCD 码，在 0001 ~ 0100（相当于 1% ~100%）之间任选，容许使用的数据区有 IO、AR、DM、HR、TC、LR 或直接用常数。图 4-23 示出 D 的含义。

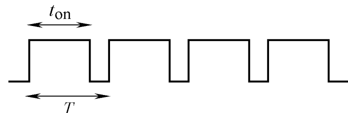


图 4-23　占空比含义

本指令为扩展指令，使用前要作功能号设定，并要下载给 PLC。对有的 PLC，在下载前，还要把 PLC 设置成容许扩展指令功能码下载模式。如 CPM2A，其 DM6602 的高字节应设为 1，否则，无法下载。容许扩展指令功能码下载的设定，也在 CXP 软件的设定窗口的“起动”表单上，选“扩展指令”为“用户设定”实行。当然，这后者实质上与前者是相同的。也是改 DM6602 的值。只是它必须下载给 PLC 后才改。

提示：DM6602 改后，PLC 还要断电，并重新上电后，这个设定才能生效。

另外，在使用这指令前，还必须在 DM 6643 的最高数位（digit）设为 1（对于用口 1），或在 DM 6644 的最高数位（digit）设为 1（对于用口 2）。如果不这么设，这两个口输出的将是未调制的脉冲。

本指令执行一次将重复输出相应脉冲。直到新的占空比的 PWM 指令执行，转而去输出新的占空比的脉冲。或到执行带参数 C = 3 的中断指令（INT），则停止输出这个脉冲。所以，本指令用微分执行也就可以了。

任何含有此功能的 PLC 也都可这么做。只是有关细节可能与此不尽相同。

用这种方法控制模拟量一般多为小型 PLC。对中、大型机较好的方法，还是用模拟量输出单元。

3. 用模拟量输出单元控制输出

为使所控制的模拟量能连续地、无波动的变化，最好的办法是用模拟量输出单元（模块）。它是把数字量转换成模拟量的 PLC 工作单元，简称 DA 单元。多数 PLC 的 DA 单元是单独的模块，但也有集成到 CPU 模块中的。

转换前的数字量可以为二进制 8 位、10 位、12 位、16 位，或更高。对应的分辨率分别为量程的 1/255、1/1023、1/4095 及 1/32767，或更小。分辨率高精度也高。

转换后的模拟量都是标准电信号——电流或电压。电流为 4 ~ 20mA。电压为 0 ~ 10V，或 1 ~ 5V，或 ± 10V 等。具体是什么，又是多少，可依型号情况及设定开关设定。

模拟量输出单元在 PLC I/O 刷新时，通过 I/O 总线接口，从总线上读出 PLC I/O 继电器或内部继电器指定通道的内容，并存于自身的内存中；再经光耦器传送到各输出电路的存储区；再分别经 D/A 转换向外或输出电流，或输出电压。

由于也用了光耦器，其抗干扰能力也很强。

DA 单元有 2 路的，还有 4 路、8 路的，少的也只有 1 路的。

有的模拟量输出单元还有一些特殊功能，即：输出限定（Out Limit）、输出限定报警（Out Limit Alarm）及脉冲输出（Pulse Output）。其含义为

（1）输出限定：可设定输出的限定使能，并设置具体的上限与下限值。有了这设定，输出将只能在这上限间变化，设定值超过上限，实际只能为上限；低过下限也类似。

（2）输出限定报警：可设定超限定具有报警的功能并设置它的相应报警值。若作了设定，则：

上限报警ON：模出≥模出限定报警上限

OFF：模出 < 模出限定报警上限 − 死区宽

下限报警 ON：模出≤模出限定报警下限

OFF：模出 > 模出限定报警下限 + 死区宽

（3）脉冲输出：可设脉冲输出使能，进而设脉冲周期及输出点。若作了设定，其脉冲

充填系数（占空比）与相应的模出量成比例。即：

$$占空比 = x/\mathrm{FFF} \times 100\%$$

这里，x 为输出通道的内容，十六进制数；

FFF 为十六进制数。

使用 DA 单元：

第一步是选用。要选性能合适的单元，既要与 PLC 的型号相当，规格、功能也要一致，而且配套的附件或装置也要选好。

第二步是接线。要按要求接线，端子上都有标明。用电压信号，只能接电压端；用电流信号只能接电流端。接线要注意屏蔽，以减少干扰。

第三步是设定。有硬设定及软设定。硬设定用 DIP 开关，软设定则用存储区，或运行相应的初始化 PLC 程序。作了设定，才能确定要使用哪些功能，选用什么样的数据转换，数据存储于什么单元等等。一句话，没有进行必要的设定，如同没有接好线一样，单元也是不能使用的。

4.2.4　模拟量执行器

执行器（actuactor）是 PLC 对受控对象施加控制作用的装置。执行器由执行机构和调节机构两部分组成：调节机构通过执行元件直接改变控制对象的参数，使控制作用满足预定的要求。

执行机构则接受来自 PLC 的控制信息，并把它转换为驱动调节机构的输出（如角位移或直线位移输出）。它也采用适当的执行元件，但要求与调节机构不同。

执行器中的执行机构，根据控制信息（电信号）产生角位移输出，以控制调节机构（如阀门的张开角度），去改变控制量（如蒸汽流量），以使调节量（如热交换器的热水出口温度）控制在所要求的数值上。

执行器直接安装在生产现场，有时工作条件严苛。特别是在被调介质具有高压、高温、剧毒、易燃、易爆、易渗透、易结晶、强腐蚀或高粘度等特点时，执行器能否保持正常工作便直接影响自动调节系统的安全性和可靠性。

执行器的类型繁多。如按执行器按所用驱动能源分，则有气动、电动和液压等多种。其相关特性，都将直接影响整个自动调节系统的调节范围和稳定性等调节品质。

为了使所编的 PLC 程序切合实际，PLC 编程人员对它也要有相应了解。

4.3　开环控制

关键词：开环特性、静态特性、动态特性、定值控制、程序控制、开环放大倍数、扰动补偿、比例控制

模拟量开环控制的类型较多，主要有：定值控制、程序控制、随动控制及补偿控制。以下先讨论开环特性。

4.3.1　开环特性

开环特性有静态特性及动态特性两种。静态特性是指，系统状态稳定后调节量（被控量）与控制量间的对应关系。而动态特性是指，系统状态在开始变化到稳定的过程中，调

节量（被控量）与控制量间的对应关系。

1. 静态特性

（1）控制量及干扰量与调节（被控）量间的量化关系。在本章开始时曾提到，调节量、控制量及干扰量。用定量分析看，它们之间关系为

$$r = f(c, t_1, t_2 \cdots)$$

这里，r 为调节量，是表示被控系统的状态、行为、性能或功能的物理量，如温度、压力、转速，等；

c 为控制量是，经 PLC 处理后、产生控制作用的输出量，如模出单元的现值，脉冲量的脉宽（占空比），开关量输出的工作时间与工作周期比等。简单的控制系统一般仅一个控制量；

t_1，t_2…是使系统的状态与行为产生所不希望变化的一些物理量，如负载电流、力矩。干扰量一般较多，但主要的可能也不多。

（2）开环放大倍数 K。调节量增量与控制量增量之比。如调节量与控制量是线性关系，则 K 为常数，否则，K 将随着控制量变化有所变化。

控制量可以是模出单元的输出值，也可是（如为脉宽调制输出）脉宽值，或别的量。依系统的构成而定。

调节量也可用模拟量输入，或脉冲量输入表示。前者，如模入单元的现值。后者，如脉冲量输入的频率。对应的也可有相应的相对的 K 值。也要按照系统构成而定。

图 4-24 所示为某电动机转速控制系统的调节量（脉冲频率，用以反映被控制电动机的转速）与控制量（模出单元输出现值，用以控制电动机转速）间的一个典型的关系。只是像这样完全线性的关系是不多的。

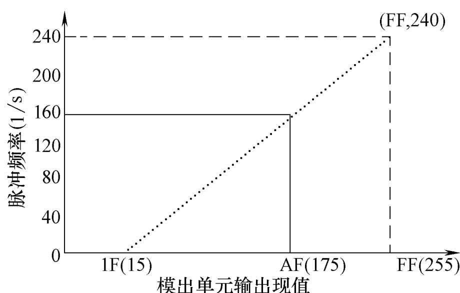


图 4-24　电动机的转速与控制量关系

这里有两点要说明：

1）PLC 模出单元有输出值，但可能系统仍无输出，即脉冲频率为 0，系统输出存在死区。如该图在 0 ~ 1F 之间时，脉冲频率均为 0。

2）输入输出关系曲线，即特性线是离散的，纵横坐标都应为整数值。

K 值反映控制系统的灵敏度。K 值大，说明控制量不大的变化，可能会产生较大的调节量的变化。

此外，系统的扰动量对调节量也会有影响。这可用开环扰动系数 T_1、T_2……表示。

开环扰动系数 T_1、T_2 指，调节量与某扰动量之比。如扰动量与控制量是线性关系，则 T_1、T_2 为常数，否则，T_1、T_2 将随扰动量不同而有所不同。

T 值反映干扰量对系统的作用强度。T 值大，说明干扰量不大的变化，可能会产生较大的调节量的变化。

但，一般讲，要弄清干扰量与调节量间的关系是不易的。

（3）输出分辨率。PLC 输出的变化是不连续的，是按输出分辨率离散变化的。如 PLC 模出单元为 12 位时，其分辨率为 1/4096。这意味着，模出变化（增或减）的最少值只能是总输出值的 1/4096，比此再小是不可能的。

与这个对应的系统输出，即调节量的变化也将是离散变化的。其变化的最小值也将是调节量最大变化量的 1/4096。

存在这个离散性说明，要使系统的调节量无误差地处于任一要求的状态是不可能的。但是，这个误差是可控制的。提高模入、模出单元的分辨率即可减少这个误差。

2. 动态特性

由于系统是开环的，输入与输出的对应关系又是惟一的，所以，开环动态特性总是稳定的。常规（线性、连续）的系统开环动态特性多是用频率特性衡量，即在其输入端人为地加上单位幅值不同频率的正弦信号，看其正弦输出的幅值变化及其与输入正弦信号的相位差。即所谓幅频特性及相频特性。

而对 PLC 控制的系统，有“误差”、“断续”及“时延”的特点，不方便用频率特性去反映它的开环特性。最简单是用响应时间，即系统从开始变化到稳定的时间衡量。具体的指标可以是：每单位控制作用，即 PLC 模出单元的每一分辨率的变化，到系统稳定所需要的时间。显然，从系统对控制作用反应的快速性讲，这个时间是越小越好。

4.3.2　开环控制

1. 定值控制

不用控制结果信息，使调节量（被控量）保持所要求定值的控制，称开环定值控制。要实现这个控制，首先要弄清控制量及干扰量与调节（被控）量间的量化关系。

这个关系已在上一节作了讨论。弄清这个关系，就可方便地进行开环定值控制了。具体的办法是，根据要求得到的调节量，依它与控制量间的对应关系，决定要输出的控制量。

这么处理的不足是，无法考虑干扰量的影响。事实上，在多数情况下，存在种种干扰是不可避免的。这样，就很难得到较好的定值控制的效果。这也是开环控制用的较少的重要原因。

2. 程序控制

这里讲的程序控制是指，使被调节量按预定规律变化所作的控制。这预定规律可根据要求任意设计。如图 4-25 所示，若运动部件起动后，先作等加速度运动；增速到 Max 值时，速度保持这 Max 值，作等速运动；到总行程“行程快到”后，作等减速度运动；减速到 Min 值后，速度保持 Min 值，作等速运动，直到行程到“行程到”时，运动停止。这就是一种程序控制规律。设计符合这样规律的控制就是这里指的程序控制。

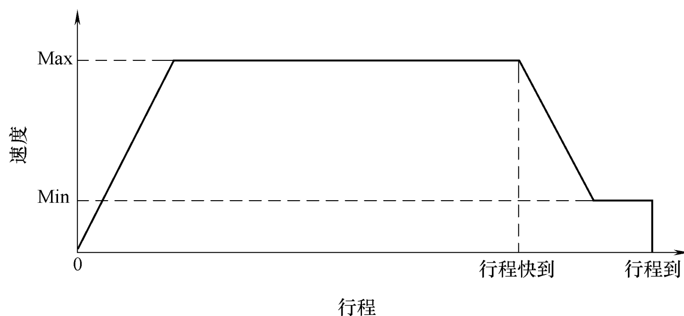


图 4-25　运动速度与行程关系

图 4-26 所示为实现这个控制的程序。当然，在使用时，还须把这里的模拟量“输出”作功率放大，以加载给直流电动机，或用“输出”控制变频器输出频率，用变频器加载给交流电动机。而且，这电动机还要去带动相应的运动部件。

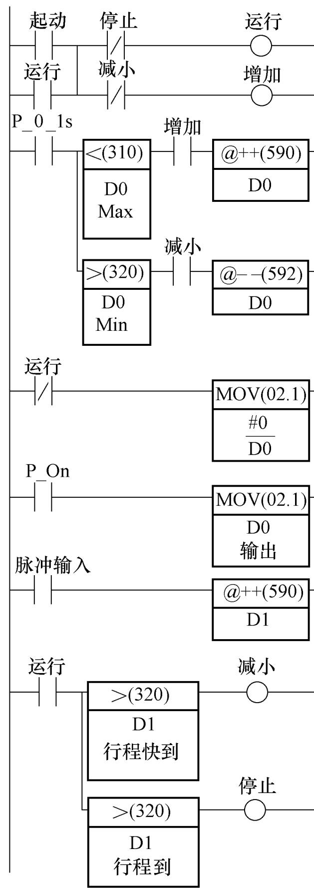


图 4-26　运动速度控制程序

图中“输出”是加载给电动机的模拟控制信号。可理解为 0 ~ 5V 或 10V 电压。此电压高，直流电压也高，或变频器的输出频率也高；用它驱动电动机时，电动机的转速也高；进而部件运动速度也快。所以，控制此数值即可控制部件运动速度。

为了便于说明，图中用的都是符号地址。如图所示，“起动”信号 ON 后，“运行”将 ON，并自保持。这时，由于“减小”OFF，其常闭触点 ON，故“增加”ON。

“增加”ON，且 D0 小于 Max 时，将使 D0 每隔 100ms 加 1 一次，十六进制数运算操作上，使数据存储器 D0 每经 100ms，加 1 一次。而此 D0 的值总是传送给“输出”，因为它的传送条件总是 P_ ON，均为常 ON 特殊继电器。这样，由于 D0 值的增大，“输出”也将随之增大，因而，所控制部件的速度将增速。

当 D0 增到 Max 值时，将保持 Max 值，不再增大。这时，部件将作等速运动。

当部件运动时，将产生“脉冲输入”信号。每输入一个脉冲，将使 D1 加 1。

当 D1 增加到等于或大于“行程快到”值时，“减小”ON，其常闭触点将使“增加”OFF。同时，使数据存储器 D0 每经 100ms，减 1 一次。这样，由于 D0 的减小，“输出”也将随之减小，因而，所控制部件的速度将减速。

当 D0 减小到 Min 值时，将保持 Min 值，不再减小。这时，部件将作等速运动。

当运动到“行程到”值时，“停止”ON，其常闭触点将使“运行”OFF。D0、D1 回到 0，整个控制完成。

应指出的是，这里的开环控制指的是速度控制。而行程控制还是闭环的。部件运动行程用脉冲输入反馈。如果运动速度较快，脉冲频率较高，还可用高速计数器处理此过程，这将在本书第 5 章还会有所讨论。

3. 比例控制

比例控制的实例如图 4-27 所示。它可使流量 Q_b 按比例 k，跟随流量 Q_a 变化。图 4-27 为它的相应程序。



图 4-27　比例控制梯形图程序

从图知，这里的“模出通道”的 BCD 码值为“三路模拟量 BCD 码”与“比例系数 K”的乘积。再经转换为十六进制码，然后输出给“模出通道”，即可使“模出通道”控

制的模拟量，按比例系数 K，随“三路模拟量 BCD 码”的变化而变化。

还可能实现多值比例控制，图 4-28 为与其对应的梯形图程序。这里有两个比例器 K1、K2，都由输入量 Qa 控制，以保证实现 Qb1 = K1 × Qa、Qb2 = K2 × Qa 的比例关系。

从图知，这里的“模出通道 1BCD 码”、“模出通道 2BCD 码”值为“三路模拟量 BCD 码”与“比例系数 K1”、“比例系数 K2”的乘积。再经转换为十六进制码后输出给“模出通道 1”、“模出通道 2”，即可使“模出通道 1”、“模出通道 2”控制的模拟量，按比例系数 K1、K2，随“三路模拟量 BCD 码”的变化而变化。

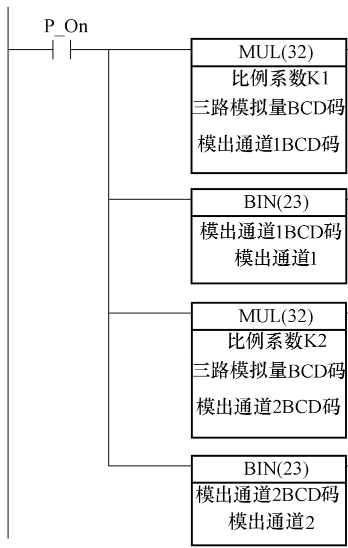


图 4-28　多值比例控制梯形图程序

提示 1：这里的乘后的“积”为双字，要确保它的“积”处在模出通道的有效值范围之内。

提示 2：如 K1、K2 不是整数，可先把 K1、K2 乘 10、或乘 100 等，使其变成整数，然后再作这里的乘。得出结果后，再用双字长除指令，把“得出结果”的除 10、或 100 等，使最后的结果处在模出通道的有效值范围之内。

4. 补偿控制

图 4-29 所示的为前馈控制，也就是这里讲的扰动补偿控制。

如该图所示，加热物料流入量就是它的主要干扰因数。如用一传感器检测热物料流入量，并通过模入单元把检测到的这个量送入 PLC，再由 PLC 按干扰规律对其进行处理（按扰动影响规律，把输入变换成相应的输出），然后再通过模出单元去控制蒸汽阀，即可实现调节蒸汽的前馈控制。从而使加热物料流入量对容器温度的干扰，得到相应补偿。

可看出，这里模入、模出、PLC 及其处理程序，即为该图的前馈补偿器。而在这几项中，最难的是弄清扰动影响规律。

一般讲，确定扰动影响规律有两个方法：解析法，探求相应函数关系；实验法，检测一系列相关数据，建对应数表。

对简单的过程，如负载电流对直流发电机的输出电压的影响，用解析法就比较好求。因为

$$U_{\mathrm{d}} = E - I_{\mathrm{d}} \times R_0$$

$$U = U_{\mathrm{d}} - \Delta I R_0 = E - (I_{\mathrm{d}} + \Delta I) R_0$$

式中　U_{d}——额定输出电压；

E——发电机电动势；

I_{d}——额定负载电流；

ΔI——实际电流与额度电流差值；

R_0——电动机电枢电阻。

可知，负载电流对输出电压的扰动是线性的。要补偿它的扰动，可提高电动势 E。而

$$E = C\Phi n$$

式中　C——电动机常数，与电动机的结构等因素有关；

Φ——激磁磁通；

n——电动机转速。

在这 3 个量中，较便于处理的是增加辅助激磁线圈，以增加的激磁磁通 $\Delta\Phi$。如用这个辅助激磁磁通 $\Delta\Phi$，所多得到的电动势 ΔE 正好等于 ΔIR_0，则可使电流变化对电压的干扰得到补偿。即：

$$\Delta\Phi = (R_0) \div (C \times n) \times \Delta I$$

有了这个关系，把用模入单元检测到的 ΔI 值，按（R_0）$\div$（$C \times n$）$\times \Delta I$ 关系，变换为 $\Delta\Phi$，并送模出单元，去产生辅助激磁磁通。即可实现这个补偿。

这里的按（R_0）$\div$（$C \times n$）$\times \Delta I$ 关系变换，对 PLC 来说，运用一些运算指令即可实现，并不难。

如图 4-6 所示的例子。弄清加热的物料流入量（扰动量）、蒸汽流量（控制量）与容器温度（调节量）的关系，从中找出解析关系，也可设计出相应的前馈补偿程序。

但是，如果找不出这些量之间解析关系，那只好用实验法。它的要点是，通过实验，逐个测出不同的扰动量时，要用多大的控制量，才能使系统的调节量达到期望值。然后，列出一个数表，存于 PLC DM 区中。

这种情况下的扰动补偿程序，就是使用好这个数表。图 4-29 即为这种程序。

从图知，这里只用两条主要指令。一是“MOV”指令，用“一路模拟量 BCD 码”对指针赋值。另一为“BIN”指令，把指针指向的 DM 地址的内容，转换为十六进制码，并传送给模出通道。与图 4-6 对照，这里的“一路模拟量 BCD 码”与“进料变送器”对应，“模出通道”与“调节阀”对应。

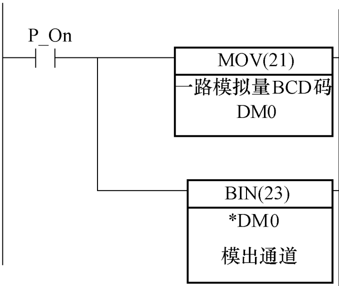


图 4-29 补偿控制梯形图程序

这里的程序较简单，但执行这程序前必须对指针 DM0 指向的 DM 区先赋值。有时，也可能从“一路模拟量 BCD 码”到指针赋值，或从指针指向的内容到“模出通道”之间，还要作一些插值运算，以使控制输出更精确些。

从上讨论可知，要设计前馈控制程序，首先要找出主要的干扰因数，并弄清扰动量、控制量与调节量的关系。否则，无法设计这种控制程序。这也是这种控制使用的局限性。

提示：比例控制及补偿控制，实际也就是随动控制，都是使一个或一组模拟量随另一模拟量的变化而变化。只是这里仍是开环的，其控制结果没有反馈给控制程序。

应指出的是，这里讲的控制输出，都是用模拟量输出单元。其实，用脉宽可调的脉冲或比例可调 ON/OFF 继电输出也是可以的。

4.4 闭环控制

关键词：反馈、反馈控制、输出 ON/OFF 控制、偏差、偏差控制、无差控制

闭环控制的目的也是多种多样的，如定值控制、随动控制、程序控制等等。其实，这 3 个控制本质上是相同的。所差的只是，定值控制的设定值是定值，而其它两个控制的设定值

是变化的。所以，以下仅对定值控制，即自动调节，进行讨论。

实现自动调节的方法有：输出 ON/OFF 控制、负反馈控制、偏差控制、无差控制等等。以下将对这些方法进行讨论。此外，还有 PID 控制、智能控制等。这些将在后续的各节中讨论。

4.4.1　ON/OFF 输出控制

这种控制的方法是，把被控量的实际值与设定值进行比较，再按照比较结果，产生相应的 ON 或 OFF 的继电控制输出。图 4-30 所示即为最简单这类梯形图程序。该程序不断执行"设定值"与"实际值"比较，只要"实际值"小于"设定值"，标志"P_ GT"ON，进而使"输出"ON；反之，"输出"将 OFF。用它即可进行输出 ON/OFF 控制。

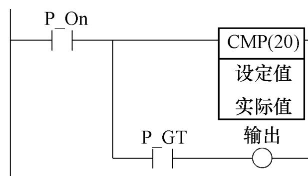


图 4-30　简单 ON/OFF 控制

这种控制只需用模拟量输入单元，而输出则用普通的 I/O 点，较简单。但可能在设定点附近 ON、OFF 动作变换过于频繁。

为避免这种 ON、OFF 动作变换过于频繁，可在比较后增加延时，再产生控制输出，或把临界点改为临界范围，比临界点大于一定数（+A）时作一种转换，比临界点小于一定数（-A）时作另一种转换。

图 4-31 所示为用上、下限比较，以产生控制输出的程序。

从图知，这个系统所控制的"实际值"低于"设定值-A"时，使其增大；高于"设定值+A"时，使其减少。

在啤酒发酵罐温度控制自动控制实例程序中，用比较加延时控制。发酵是啤酒生产的重要过程，占的时间也最长（约一个月）。发酵过程会自动升温，但温度过高会使啤酒变质。故要经常向罐中冷却管道注入冷气，以保持罐不超温。但，注入冷气太多罐温度太低也不行。那样，将影响发酵进程。故须在啤酒发酵过程中，对其温度进行控制。

图 4-32 为其梯形图程序。图中 110 通道的内容为处理后的输入温度值，而 DM981 的内

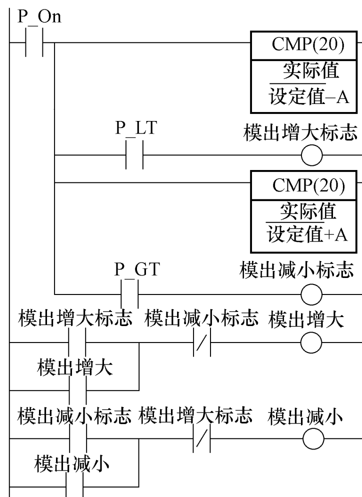


图 4-31　上限下限比较控制程序

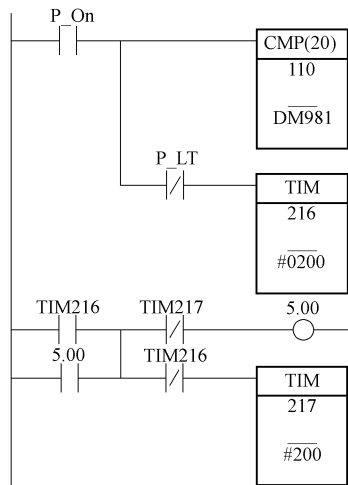


图 4-32　延时 ON/OFF 控制

容为温度设定值。从图知，当输入值大或等于设定值，并超过 20s，则可使 00500 ON 产生输出，打开冷气阀门，向冷却系统放入冷气。若温度低于设定值，并超过 20s，则 00500 OFF，阀门关闭，停止放冷气。

这里设的延时为 20s，实际可依系统的惯性调整。这里适当的延时可避免干扰对系统的影响。

此外，啤酒发酵工艺，要求罐的温度要逐步下降，直到某个值后，再保持恒温。所以，在事实上，它的控制是程序控制。所以，图 4-32 中的 DM981 的内容是随时间变化，即给设定值不是常数。

图 4-33 是使 DM981 变化的梯形图程序。这里用的是随时间作线性减小变化。总的时间范围设在 HR50 中，最大不大于 9999min，最小不能为零。这是在运行这段程序前确定好了的。变化范围高位值放在 HR20 中，低位值放在 HR10 中。

提示：这里可把 HR50 直接作为 CNTR 设定值的地址，也可实现 CNTR 计数最大值的控制，但如中途改变 HR50 的值，此改变值计数器须复位后才起作用。 而现在的处理，HR50 的值随时改，随时就起作用。

这里先进行减运算，把 HR20 被 HR10 减，其结果存于 HR31 中。然后作双字除运算。OMRON C200H 机的除指令，只能进行整除，不计小数。故这里用的被除数为 HR30，其含义是 HR31、30 作被除数（HR30 为零），而除数为 HR50 及 HR51（HR51 为零）。其商存于 HR40 及 HR41 中。

接着进行乘，这里用 HR41 与 C000 的现值相乘。因在进行除时，被除数放大 10000 倍，而在乘时，又缩小 10000 倍（HR40 不用），两者相抵，等于不放大，也不缩小。乘后的积再与 HR20 相减，并存于 DM981（低 4 位）及 DM982（高 4 位）中。但 DM982 是恒等于零的。而 D981 正是我们所需要的。

提示：由于用的是 PLC 整除指令，若温差（HR31）值小于时间范围（HR50）值，相除后，商肯定为零，……那是不可能使 DM981 随时间作减的变化的。

上述计算程序，用于沈阳华润啤酒厂的发酵罐温度控制，经多年使用，效果甚佳。

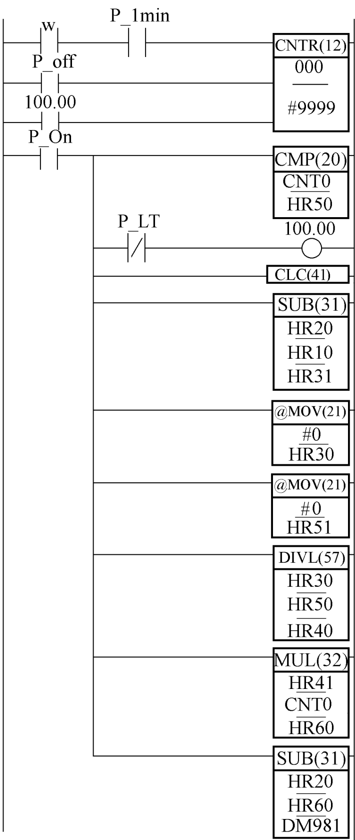


图 4-33　给定值变化的自动控制

4.4.2　负反馈控制

图 4-34 表示了负反馈控制的算法框图。这里是用设定值（R）与实际值（C）之差（E）去进行控制。E 经放大（乘 H）后，产生控制输出 M_1，作用于

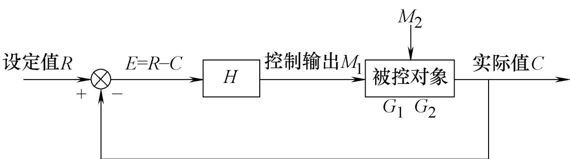


图 4-34　负反馈控制算法框图

被控对象。M_2 为某干扰量，不可避免地也同时作用于被控对象。被控对象的 G_1、G_2 为接受控制与扰动的特性值，是由系统的特性决定的。即 $C = M_1G_1 + M_2G_2$。

从框图知，以上几个量间还有如下关系：

$$E = R - C$$

$$M_1 = HE = H\ (R - C)$$

以上关系，经化简后有：

$$C = \frac{HG_1R + G_2M_2}{1 + HG_1}$$

如果 HG_1 选得足够大，既足够的大于 G_2，又足够的大于 1，则有：

$$C = R + \frac{G_2}{HG_1}M_2 \approx R$$

这可使 C 不受或很少受干扰影响，而复现 R 的变化，达到精确控制的目的。

提示：这里及以下各小节的计算都是静（稳）态下的计算。没有考虑部件的动力学特性及受其影响产生的过渡过程。精确计算要使用传递函数，请参阅自控专著，或作者的《PLC 硬件配置及软件编程》一书。

当然，这里讲的只是理想情况。而实际系统是离散的，不仅系统总有惯性，而且 PLC 控制还都有滞后（时延），所以，HG_1 选得足够大，虽可改善系统的静态特性，但，很可能出现静态及动态不稳定（振荡）。这也是负反馈控制的不足。

建立这个系统后，可按所要求的被控量 C 值，确定控制量 R 值，以实现相应的定值、或程序、或随动控制。

图 4-35 所示的与图 4-34 对应的梯形图程序。该图用的是符号地址。

从图知，当“负反馈控制”ON，则执行如图所示程序。这时，先用“CLC”指令清进位位（P_CY），接着，进行“设定值”与“实际值”相减，结果存于“E”中。如“设定值”大于“实际值”，则 P_CY OFF，“E”即为这个差值，进而把“E”乘“H”，并存于“控制输出”中。如“设定值”小于“实际值”，则 P_CY ON，接着，把 0 传送给“控制输出”。“设定值”小于“实际值”时作这么处理，目的是避免“控制输出”出现负值。如“控制输出”允许出现负值时，则更简单。把“设定值”与“实际值”相减得出“E”（或正、或负），再将“E”乘“H”，并把结果存“控制输出”即可。但，这时 BCD 码运算要使用带符号的数据。

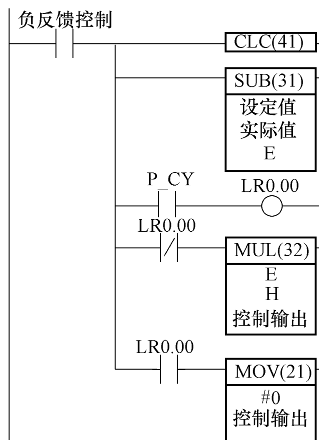


图 4-35　负反馈控制梯形图程序

提示：编写这类程序一定要处理好数据格式的转换。这里的数据运算指令用的数据格式为 BCD 码。而模拟量入、模拟量出单元用的数据格式多为十六进制数。所以，在其间要进行转换。新型 PLC 运算用的数据格式可为浮点数，用它可以提高运算精度。但从模入单元读的数到进行计算，把此计算的结果送模出单元，也都要进行格式转换。

4.4.3 偏差控制

图 4-36 表示了偏差控制的算法框图。这里用控制值 R 对系统进行控制，而设定值为 R_0。当设定值（R_0）与实际值（C）之差（E）为 0 时，R 即等于 R_0。这等于实际值复现了设定值。如由于干扰，实际值偏离设定值，则 E 不等于 0。此时，E 将增大 R 值（当实际值小于设定值时），或减少 R 值（当实际值大于设定值时），进而使实际值（C）增大，或减少，以接近设定值。

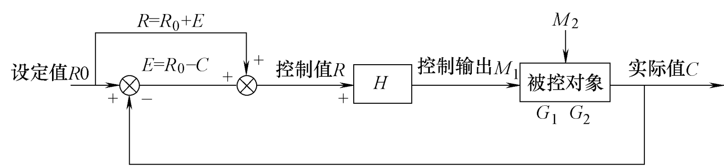


图 4-36 偏差控制算法框图

图 4-37 为与图 4-36 对应的梯形图程序。

从图知，当“偏差控制”ON，则执行如图所示程序。这时，先用“CLC”指令清进位位（P_CY），接着，进行“设定值”与“实际值”相减，结果存于“E”中。如“设定值”大于“实际值”，则 P_CY OFF（触点 LR0.00 也为 OFF），“E”即为这个差值。进而把“E”与“设定值 R_0”相加，再乘“H”，并存于“控制输出”中。如“设定值”小于“实际值”，则 P_CY ON（触点 LR0.00 也为 ON），接着，取“E”的补数，进而把“E”与“设定值 R_0”相减，再乘“H”，也存于“控制输出”中。当然，这是如“E”大于“设定值 R_0”，则把 0 传送给“控制输出”。“设定值”小于“E”时作这么处理，目的是避免“控制输出”出现负值。

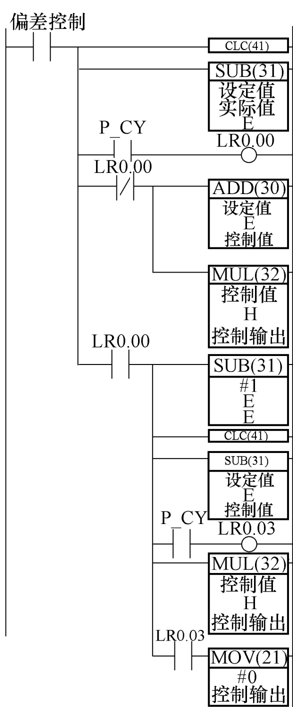


图 4-37 偏差控制梯形图程序

偏差控制实质也是负反馈控制，不同的只是这里有由设定值 R_0 确定的初始实际值。只要这个实际值发生了偏差，将产生的补充控制量，使系统的实际值恢复初始实际值。所谓偏差控制，也因此得名。

但是，要想用这种偏差控制去完全消除是不可能的。因为没有偏差也就没有补充控制量，怎么把所有偏差都消除了。只是加大 H 值，使这个无法消除的偏差，也叫静差减小。可是，这么做也存在系统是否稳定的问题。所以，加大 H 是受限制的。

提示：OMRON 加、减指令执行时，其进位位也参与运算。如图 4-35，用 SUB 指令算时，进（此时实为借位）位位置 1，差为 9999。求其绝对差，则用 0~9999 即可。但图 4-37 用了 1~9999。原因是，这时进位位已置 1。

4.4.4 无静差控制

图 4-38 表示了无静差控制的算法框图。这种控制可完全消除静差。与图 4-36 不同的只

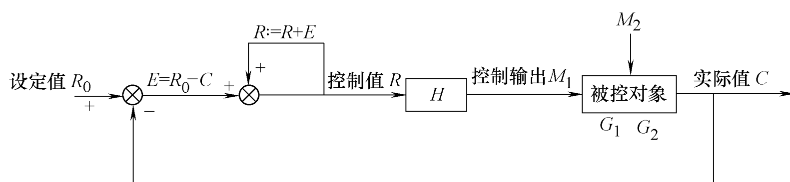


图 4-38　无静差控制算法框图

是，这里的控制值不是由偏差 E 与“设定值”相加产生，而是在每进行一次此类运算时，自身与偏差 E 相加。这样，可实现即使无偏差，但 R 可以大于设定值为 R_0，实现补充控制。

图 4-39 为与图 4-38 对应的梯形图程序。

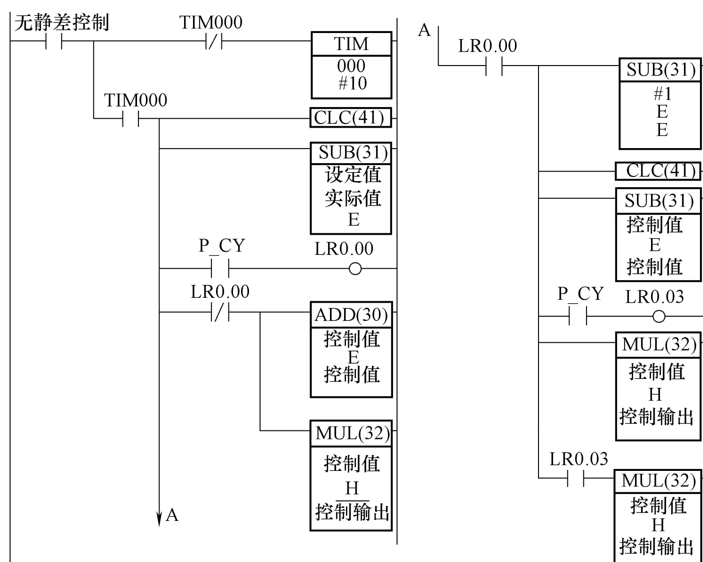


图 4-39　无静差控制梯形图程序

从图知，当“无静差控制”ON，则执行如图所示程序。先是定时器 TIM 000 工作。每 1s，其常开触点 TIM 000 ON 一次，即本控制程序执行一次。这么做的目的是，控制输出改变后要等待一定时间再处理，以适应系统时延特性。否则会出现超调，以至于系统振荡。这里定时间隔为 1s，如系统时延较大，还可加大。本程序与图 4-37 程序不同的只是，用 E 与“控制值”累加（或累减），代替 E 与“设定值”的加（或减）。

由于这里用了累加、累减，相当于加入积分环节。所以，只要存在偏差，即“设定值”减“实际值”不为 0，那么，每进行一次运算都将使“控制值”变化。直到“控制值”变化到使系统实际值等于设定值时，即“设定值”减“实际值”为 0，不存在偏差时，“控制值”才保持不变。这也就实现了设定值对系统的无静差控制。

无静差控制可消除所有由干扰产生的误差，这是它的优点。但同样存在系统能否稳定工作的问题。而系统要是不稳定，那即使静态精度再高，也是不允许的。所以，还得寻找更好控制办法。以下将介绍的 PID 控制就是一个较好的办法。

提示 1：闭环控制振荡有两种原因：静态原因及动态原因。

提示 2：静态原因是系统离散性。系统离散，就不可能任意要求的调节量都存在。如设定值正是这个不存在调节量，那是永远做不到的。而硬要这么做，只好振荡了，即出现调节量一会儿比设定值大，一会儿比设定值小，总也稳定不下来。正如，要买 1 斤鸡蛋，很难做到买的正好是一斤，要不稍多一点，要不就是稍差一点。显然，提高分辨率（如鸡蛋很小）可减少这个振荡。

提示 3：动态原因是系统惯性、时延。前者在实际的系统中也总是存在的。减少惯性，减少时延，选择合适放大倍数，可避免出现振荡。

4.5 PID 控制

关键词：比例（P）、积分（I）、微分（D）、偏差、积分常数、微分常数、放大倍数（比例系数）

模拟量闭环控制较好的方法之一是 PID 控制。它作为最早实用化的控制已有 60 多年历史，现在仍然是应用最广泛的工业控制。PID 控制简单易懂，使用中不必弄清系统的数学模型。有人称赞它是控制领域的常青树是不无道理的。

4.5.1 PID 控制基本公式

PID 是比例（P）、积分（I）、微分（D）之意。标准 PID 的控制值是与偏差（设定值与实际值之差）、偏差对时间的积分、偏差对时间的微分，三者之和成正比。如用式子表示，即：

$$p = K\left(e + \frac{1}{T_i}\int_0^t e\mathrm{d}t + T_d\frac{\mathrm{d}e}{\mathrm{d}t}\right) + M \tag{4-1}$$

式中 p——控制值；

e——偏差；

T_i——积分常数；

T_d——微分常数；

K——放大倍数（比例系数）；

M——偏差为零时的控制值，有积分环节存在此项也可不加。

PID 控制就是用这里的控制量 p，去进行对控制对象的控制。可知，它要使用反馈信号，所以闭环控制。这里的偏差、偏差对时间的积分、偏差对时间的微分，又分别称为比例作用、积分作用和微分作用。

式（4-1）用于连续系统的 PID 控制。如在 PLC 控制中用它，则必须将其“离散化”，用相应的数值计算，代替这里的积分、微分。

如选择的采样周期为 T，积分初值为 0，离散化后的式（4-1）为

$$p(n) = K\left\{e(n) + \frac{T}{T_i}\sum_{j=0}^{n} e(j) + \frac{T_d}{T}[e(n) - e(n-1)]\right\} + M \tag{4-2}$$

式（4-2）的计算仅仅是加、减、乘、除等基本运算。所以，如选定了采样周期 T、积分常数 T_i、微分常数 T_d、放大倍数 K、偏差为零时的控制值 M 以及各个时刻的偏差值，用 PLC 的算术运算指令完全可进行这个运算，以求出不同时刻 n 的控制值。进而再把这个控制

值输出，即可实现 PID 控制。

PID 控制用途广泛、使用灵活，已有很多成功的实例。在使用时，只需设定好 3 个参数（K_p、K_i 和 K_d）即可。在很多情况下，并不一定需要全部 3 个控制，可以取其中的一或两个控制，但比例控制控制是必不可少的。

虽然很多工业过程是非线性或时变的，但通过对其简化可以变成基本线性和动态特性不随时间变化的系统，这样 PID 控制就可以用了。

其次，不少 PLC 已具有 PID 指令、PID 函数块、PID 单元或回路控制单元，为 PLC 使用 PID 控制，提供了方便。

PID 参数 K_p、K_i 和 K_d 可以根据过程的动态特性及时整定，以达到较好的 PID 控制效果。如果过程的动态特性变化，例如可能由负载的变化引起系统动态特性变化，PID 参数就可以重新整定。而且，新型的 PLC，如 OMRON CJ1 机，其 PID 指令执行时，随时都可对 PID 参数进行自整定，为确定 PID 参数提供了很大方便。

但是，PID 控制也有其局限性。PID 在控制非线性、时变、耦合及参数和结构不确定的复杂过程时，效果不是太好。如果系统过于复杂，有时可能无论怎么调参数，都不易达到目的。

4.5.2　PID 控制参数含义

要是 PID 控制取得好的效果，关键是选定 PID 控制参数。要选定这些参数，首先要是对这些参数的物理意义有所了解。

参数 M，偏差为零时的控制值，即系统平衡时所要求的控制输出。可依系统静态特性确定。

参数 T 采样周期，即计算式（4-2）的时间间隔。此值太大，式（4-2）将无法正确地反映式（4-1）。太小，则增加计算工作量，影响控制的即时性。为了不失真，依采样定理要求，它的倒数，即采样频率一般应大或等于系统最大频率的两倍。困难的是这个系统最大频率也不好确定。在无法确定系统最大频率的情况下，应尽可能取得小些。

参数 T_i，积分常数，此值小，积分作用增强，但有可能影响系统的稳定；太大，稳定性好，但积分作用弱。有否积分作用，不仅决定于当前偏差存在与否，还与偏差的历史有关。正是有了这个积分作用，才能消除静差。

参数 T_d，微分常数，此值大，将增强微分作用；小，微分作用弱。微分作用加入，使得偏差刚产生，或增大时，增强控制作用，可加速缩小偏差；而偏差减少时，将减弱控制作用，可防止超调，增加系统的稳定性。虽然如此，但不是说微分作用越强越好。事实上，微分作用太强，也不利于噪声干扰的抑制，也会引起相应的振荡。有时，微分作用还被众多控制工程师认为是缺陷多于优点。因此，常要对太强的微分作用加以限制。

参数 k，比例作用，也即整个控制作用的放大倍数。此值大，将增强偏差的控制作用，减少控制误差。但有可能影响系统的稳定，以至于使系统出现振荡；太小，稳定性好，但控制作用弱，减少控制误差的时间将加长。比例作用，有的用比例带（或比例度）δ 表示，δ 与放大系数 k 的关系为倒数关系。即：

$$\delta = 1/k$$

PID 三项可作组合，简单系统可仅用 P，复杂一点的可用 PI，再复杂的这三项可都用。

PID 作用由比例作用（P），积分作用（I）和微分作用（D）共同组成。

图 4-40 分别示出，当设定值从 0 突变到 x 时，在比例（P）作用、比例积分（PI）作用和比例积分微分（PID）作用下，被调量 y 变化的过渡过程。可以看出，比例积分微分作用效果为最佳，能迅速地使 y 达到设定值 x。比例积分作用则需要稍长的时间。比例作用则最终达不到设定值，而有余差。

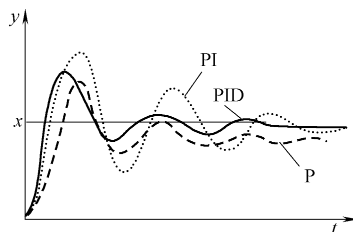


图 4-40 控制特性

4.5.3 PID 控制算法设计

图 4-41 表示了 PID 控制的算法框图。从图知，它的控制值 R 是 I、E_n、D 三部分的和。控制输出，则是 R 乘以比例系数 H。这里未计及偏差为零时的控制值 M，但这可通过执行积分运算自然实现。

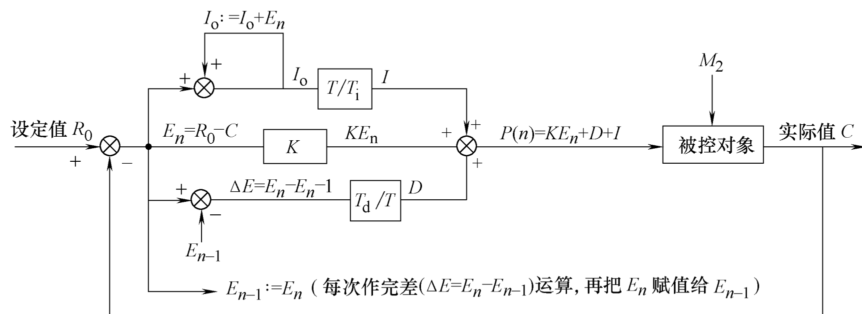


图 4-41 PID 控制的算法框图

4.5.4 PID 控制程序实现

1. 图形图程序

图 4-42 为与图 4-41 框图算法对应的一种梯形图。它用的是符号地址，较便于理解。

图上标有 18 个注解，分别说明如下：

① 用定时器 TIM001 实现每一秒（此值可设）执行一次 PID 运算。这里一秒即为采样周期 T。

② 求设定值与实际值之差。

③ 如“设定值”比“实际值”小，则先求偏差的绝对值（因借位位也参与运算，故加了 1）。然后，用它去减“积分值 0”。

④ 如“积分值”0 小于 0 ，则其取值为 0（最小值控制）。

⑤ 如“设定值”比“实际值”大，则把偏差“En”加“积分值 0”。

⑥ 如加后“积分值 0”超过 9999，则其取值为 9999。

⑦“积分值 0”除积分常数，其商存于“积分值”。

⑧ 求偏差的变化⑧a保存原偏差值。

⑨ 如偏差为负变化，则求其补码。

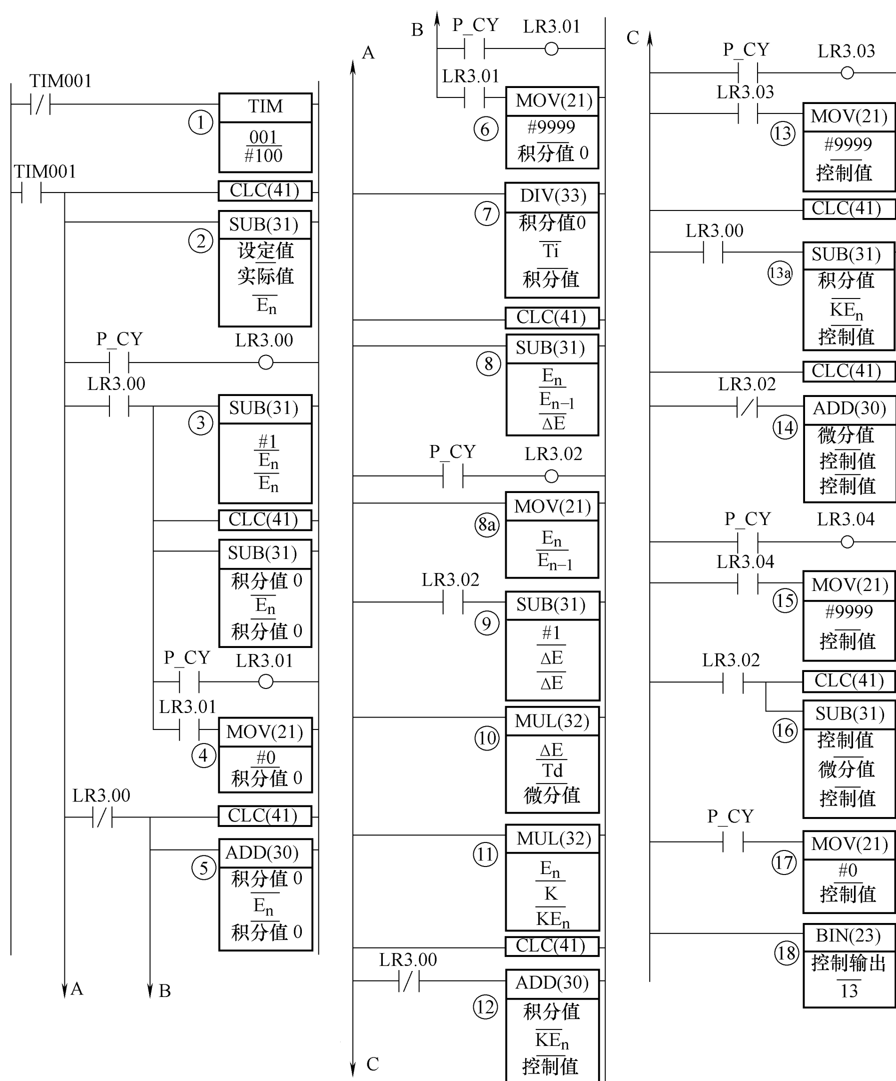


图 4-42　PID 控制梯形图程序

⑩ 偏差的变化乘微分常数。

⑪ 偏差值“En”与比例系数“K”相乘，其积存于“KEn”。

⑫ 如“设定值”比“实际值”大，则“积分值”与“KEn”偏差相加，其和存于“控制值”。

⑬ 如加后有进位，即大于 9999，则“控制值”取值为 9999。

⑬a 如“设定值”比“实际值”小，相减。其后如需要也可作最小值控制。

⑭ 如偏差为正变化“微分值”与“控制值”相加，并存于“控制值”

⑮ 如加后有进位，即大于 9999，则“控制值”取值为 9999。

⑯ 如偏差为负变化，则“控制值”与“微分值”相减，并存于“控制值”。

⑰ 如“控制值”小于 0，则其取值为 0。

⑱ 把“控制值”转换为二进制数，并送通道 13，即模拟量输出通道。

提示：如果这里的数据格式为单字长 BCD 码，最好用双字长，运算指令也用双字长。特别存储积分结果值的数据格式，尽量用双字长。 这样，可提高控制精度，而且也便于 PID 参数整定。

2. ST 语言程序

PID 控制使用的数学运算较多。所以，使用 ST 语言编程更简便些。以下为它的 ST 语言程序。

(1) 变量声明

1) 输入变量。有：sdZI（设定值），INT；sjZI（实际值），INT；jfCS（积分常数），REAL；wfCS（微分常数），REAL；blCS（比例系数），REAL；jfZI（上次积分值入），REAL；sjZI0（上次实际值入），REAL。

2) 内部变量。有：pCZN，REAL；wfZN，REAL；sdZN，REAL；sjZN，REAL；jfZN，REAL；kzZN，REAL。

3) 输出变量。有：kzO（控制输出），INT；sjZO（本次实际值保留），REAL；jfZO（本上次积分值出），REAL。后两个变量应作为既是输出，又是输入处理。新版本的 ST 语言可定义为输入输送出变量。那样，这里的 jfZO 与 jfZI 就是一个变量，sjIO 与 sjZO 就是一个变量。

(2) ST 语言语句。含义见注解。显然，这里的程序比梯形图程序要简明得多！

```
sdZN:=INT_TO_REAL(sdZI);(*设定值转换为实数*)
sjZN:=INT_TO_REAL(sjZI);(*实际值转换为实数*)
pCZN:=sdZN-sjZN;(*计算偏差*)
jfZN:=jfZN+jfCS*pCZN;(*计算积分分量*)
wfZN:=wfCS*(sjZN-sjZIO);(*计算微分分量*)
kzZN:=blCS*pCZN+jfZN+wfZN;(*计算控制输出*)
kzO:=REAL_TO_INT(kzZN);(*控制输出转换为整型数*)
sjZO:=sjZN;(*保留实际现值,为微分计算作准备*)
```

(3) 功能块调用。ST 编写的功能块，使其工作还得有调用主程序。图 4-43 即为它的调用程序。只是这里模拟量输入、输出没有表示。

这里定时器 TIM 0001 设定的时间为 PID 运算的时间间隔。D0 ~ D16 为输入。D0 存储的为设定值，D4 存储的为实际值，D8 存储的为积分常数，D12 存储的为微分常数，D16 存储的为比例系数。D20 存储的为积分值，D24 存储的为实际值。而 D28 为控制输出。D20、D24 为既是输入，又是输出。用以存储运算的中间结果。

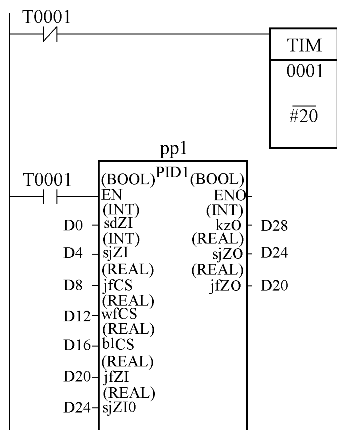


图 4-43 功能块调用程序

4.5.5 PID 控制参数选定

从以上介绍知，实现 PID 控制，对有关参数作合理整定是重要的。PID 控制参数，多是先确定采样周期 T，再就是比例系数 K，然后为积分常数 T_i，再就是微分常数 T_d。

而且这些参数整定，多都是在凭经验，在现场调试中具体确定。一般是，先取一组数据，将系统投运，然

后对系统人为加一定扰动，如改变设定值，再观察调节量的变化过程。若得不到满意的性能，则重选一组数据。反复调试，直到满意为止。

大体的步骤是：

1. 选定合适的采样周期 T

对流量控制系统，一般为 1～5s，优先选用 1～2s。

对压力控制系统，一般为 3～10s，优先选用 6～8s。

对液位控制系统，一般为 6～8s。

对温度控制系统，一般为 10～20s。

对成分控制系统，一般为 15～20s。

当然，以上这些数据也是很笼统的，仅仅是参考数。

从实际经验看，T 最好尽可能选得小些。T 小，并把积分作用（见以下参数 T_d 解释）适当减弱，可做到，既加快调节过程，而又避免由系统离散原因引起的超调。

2. 选定合适的比例带 δ

这时设 T_i 为∞（无穷大），T_d 为 0，从大到小改变比例带，直到得到较好的的过程曲线。

3. 选定合适的积分常数 T_i

将比例带放大 1.2 倍，从大到小改变积分时间常数，直到得到较好的的过程曲线。

4. 再定合适的比例带 δ

积分时间常数不变，再改变比例带（增大或减小），看过程曲线是否改善。若有改善，则继续调整比例带；若没有改善，则将原定的比例带减小，再变更积分时间常数，以改善控制过程曲线。如此多次反复，直到得到合适的比例带及积分时间常数。

5. 选定合适的微分常数 T_d

一般讲，微分时间常数可在积分时间常数的 1/6～1/4 间选定。而且，引入微分作用后，积分时间常数可适当减小。这些参数选定后，再观察过程曲线是否理想。如不当，可再作相应调整，直到满意为止。

有关 PID 控制器（硬件 PID 控制器）的 PID 参数整定，曾流行一些口诀，现转载介绍如下，供参考！

参数整定找最佳，从小到大顺序查。
先是比例后积分，最后再把微分加。
曲线振荡很频繁，比例度盘要放大。
曲线漂浮绕大湾，比例度盘往小扳。
曲线偏离回复慢，积分时间往下降。
曲线波动周期长，积分时间再加长。
曲线振荡频率快，先把微分降下来。
动差大来波动慢，微分时间应加长。
理想曲线两个波，前高后低 4 比 1。
一看二调多分析，调节质量不会低。

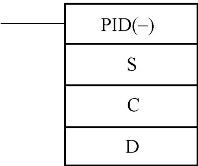
4.6　PID 指令及其应用

关键词：PID 指令、积分常数、微分常数、比例带、两个自由度 PID 控制

目前，多数 PLC，如 OMRON 公司的多种 PLC，都有 PID 指令。所以。用 PID 指令实现模拟量的 PID 控制是相当方便的。

4.6.1 PID 指令说明

一般讲，OMRON 公司 PLC 的 PID 指令格式是：



对 C200HS、CPM2 *、CPM2 *-S *、SRM-V2、CQM1（CPU4x 仅仅是），CQM1H 和 C200HX 系列 PLC，PID 为扩展功能指令，使用前，其功能码应先作分配。而 CS1G、CJ1G、CS1H、CS1G-H、CS1H-H、CJ1G-H、CJ1H-H 及 CJ1M 机，其 PID 指令功能码为 190。

这里：

S 为输入字，十六进制码，调节量（实际值）输入到这个字中。

D 为控制输出字（通道），十六进制码，运算后的 PID 值，即控制值，放在这个字中。

C 为 PID 控制参数首字，从 C1 ~ C1 + 32，这 33 个字应放在同一连续数据区内。C200HS 等机型 C1 ~ C1 + 32 的各个字的含义见表 4-1。而 CS1G 及其后续机，增加了输出上下限控制，其参数共 39 个字，也应分布在同一连续的数据区中，其中 C1 + 7、C1 + 8 即用于存所期望的输出下限值及上限值。表 4-2 所示的就是它的各参数含义。从中可知，有关参数是略有差别的。

表 4-1 控制字含义

字	位	参 数 名	功能/设定范围
C1	00 ~ 15	设定值(SV)	PID 控制目标值,二进制,位数同输入
C1 + 1	00 ~ 15	比例带宽	比例带宽与输入范围之比,BCD 码,0000 ~ 9999 之间,对应于 0.1% ~ 999.9%
C1 + 2	00 ~ 15	积分时间	用于积分控制,积分时间与采样时间之比,BCD 码,0000 ~ 9999 之间,对应于 0.1% ~ 999.9%（9999 时积分不起作用）
C1 + 3	00 ~ 15	微分时间	用于微分控制,微分时间与采样时间之比,BCD 码 0001 ~ 8191 之间,或 0000
C1 + 4	00 ~ 15	采样周期	采样周期,BCD 码,0001 ~ 1023,对应时间为 0.1 ~ 102.3s
C1 + 5	00 ~ 03	PID 正反向控制指定	0 反向控制,1 正向控制
	04 ~ 15	输入滤波系数	确定滤波强度,系数小、滤波弱,BCD 码,100 ~ 199,或 000,对应于 0.00 ~ 0.99 默认值为 0.65

（续）

字	位	参 数 名	功能/设定范围
C1 +6	00 ~07	输出范围	确定输出数据二进制位数,00 ~08 对应的输出位是 8 ~16 位
	08 ~15	输入范围	确定输入数据二进制位数,00 ~08 对应的输入位是 8 ~16 位
C1 +7 ~C1 +32	00 ~15	工作区	系统留用

表 4-2　控制字含义续

控制数据	项　　目	内　　容	设 定 范 围	置 ON 输入条件变化
C	设定值(SV)	控制过程的目标值	二进制数据(与指定输入范围位数相同)	不允许
C +1	比例带	表示比例控制范围/总控制范围的 P 作用参数	0001 ~270FH(1 ~9999) (0.1% ~999.9%,以 0.1% 为单位)	如果 C +5 的位 1 是 1,输入条件 ON 可以改变
C +2	Tik	表示积分作用强度的一个常量,数值增加,积分强度减小	0001 ~1FFFH(1 ~8191) (9999 = 不执行积分操作)①	
C +3	Tdk	表示微分作用强度的一个常量,数值增加,微分强度减小	0001 ~1FFFH(1 ~8191): (0000 = 不执行微分操作)①	
C +4	采样周期	设定执行 PID 作用的周期	0001 ~270FH(1 ~9999) (0.01 ~99.99s,以 10ms 为单位)	不允许
C +5 中位 04 ~15	2-PID 参数(α)	输入滤波系数,通常为 0.65(即 000 设定),过滤效率随系数向 0 靠近而递减	000H:α =0.65 设置从 100 ~163H 表示最右边的两位的值被设定为从 α = 0.00 ~α =0.99②	
C +5 中位 03	操作变量输出设定	设定 PV 等于 SV 时的输出变量控制	0:输出 0% 1:输出 50%	
C +5 中位 01	PID 常量变化设定	在 PID 计算中,可以改变比例带(P),积分常数(Tik)与微分常数(Tdk)的时刻设定	0:PID 指令开始执行 1:PID 指令开始执行和每一个采样周期	允许
C +5 (位 00)	PID 前向/逆向设定	决定比例作用的方向	0:反向 1:正向	不允许
C +6 (位 12)	操作变量输出限位控制	决定输出控制变量是否应用限位操作	0:无效(无限位控制) 1:有效(有限位控制)	
C +6 (位 08 ~11)	输入范围	输入数据位数	0:8 位　5:13 位 1:9 位　6:14 位 2:10 位　7:15 位 3:11 位　8:16 位 4:12 位	

（续）

控制数据	项　　目	内　　容	设 定 范 围	置 ON 输入条件变化
C +6（位 04 ~07）	积分和微分单位	决定积分和微分常量的单位	1：采样周期倍数 9：时间（单位：100ms）	不允许
C +6（位 00 ~03）	输出范围	输出数据位数（输出位数自动等于输入位数）	0：8 位　5：13 位 1：9 位　6：14 位 2：10 位　7：15 位 3：11 位　8：16 位 4：12 位	
C +7	输出变量下限	操作变量输出限位有效时的下限	0000 ~ FFFF（二进制）③	
C +8	输出变量上限	操作变量输出限位有效时的上限	0000 ~ FFFF（二进制）③	

① 单位设为 1 时，范围为从 1 ~8191 倍周期。单位设定为 9 时，范围为从 0. 1 ~819. 1s。设为 9 时，将积分和微分时间设在 1 ~8191 倍采样周期的范围内。

② 设定 2-PID 参数（α）为 000（0. 65），即标准值。

③ 当输出控制变量限位控制有效（即设为 1）时，设定如下值：输出范围最大值≥MV 输出上限≥MV 输出下限≥0000。

4. 6. 2　两个自由度 PID 控制

OMRON PLC 的 PID 指令还用了两个自由度的目标值 PID 控制技术。两个自由度的目标值 PID 控制技术的特点是，除了反馈控制，还用了前馈控制。具体的控制框图如图 4-44 所示。

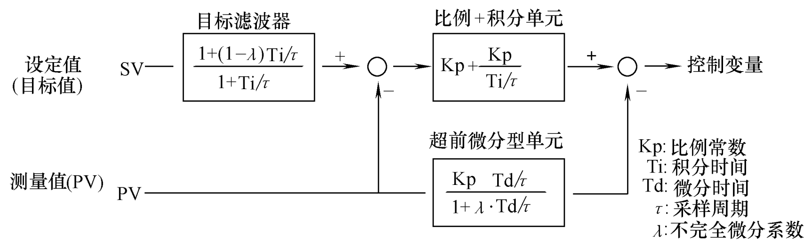


图 4-44　两个自由度算法框图

从图知，它的控制值不仅取决于对偏差值的 PID 运算，而且，还对当设定值作了滤波处理，设定值的变化，也会提前（前馈）作用于控制变量。

有了这个设定值变化的前馈处理，可改善在设定值改变时，系统的动态特性。这个技术是 OMRON 专利，不仅其 PLC 的 PID 控制用它，而且，其温控仪表也用它。

提示：怎么使用这个前馈控制，要对控制字 C +5 的第 4~15 位作相应设定，以确定这个前馈作用的强弱。默认为 000，其滤波系数为 0. 65。

4. 6. 3　PID 参数选定

使用 PID 指令实现模拟量 PID 控制是很方便的，按上述要求调用这个指令就可以了。但

还要选定好有关的 PID 参数，以确保所作的 PID 控制有较高的控制品质。

这些参数与对应控制控制作用相关。如比例带与偏差控制作用相关；积分常数与积分作用相关；微分常数与微分作用相关。此外，最重要的一个是采样周期。

1. 采样周期

采样周期，是执行 PID 运算的的时间间隔。此值太大，计算进程减缓，积分的作用将减弱，并影响控制的即时性。太小，则相反。至于怎么选取，在上一节已作了介绍。

提示：实际经验看，*T* 最好尽可能选得小些。*T* 小，并把积分作用适当减弱，可做到，既加快调节过程，而又避免超调。

要指出的是，这里设的采样周期，并不一定就是实际的采样周期。在随后，还要对其做讨论。

2. 比例带

比例带与以上介绍的比例常数类似，但含义相反。它的大小与比例作用有关。比例带小，即比例常数大，比例作用强，反之，弱。如图 4-45 所示（图 4-45a 为反向控制，即实际值大时，控制输出减小，图 4-45b 为正向控制，即实际值小时，控制输出增大），当实际值（PV）小于比例带，控制输出（MV）为 100%，即最大值。在比例带内，MV 与偏差（设定值 SV 与实际值 PV 之差）成比例并逐渐减小，直至 SV 与 PV 匹配（即直到偏差为 0），此时 MV 为最小值 0%（或 50%，根据控制变量输出定义参数的设定而定）。当 PV 大于 SV 时，MV 也为 0%。

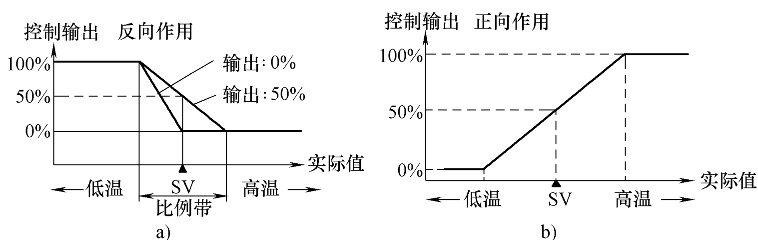


图 4-45　正、反控制

比例带越小，比例常数越大，控制作用也越强。同时，减小比例带还可可减少静差，但比例带太小则易产生振荡。比例带变化对控制的特性影响如图 4-46 所示。

3. 积分作用

积分作用是加速控制进程，同时可消除静差。此常数小，积分作用增强；大，相反。积分作用太强，可能出现宽幅振荡、超调和欠调。此时，如果增加积分时间或扩大比例带，则可减少振荡，如图 4-47 所示。

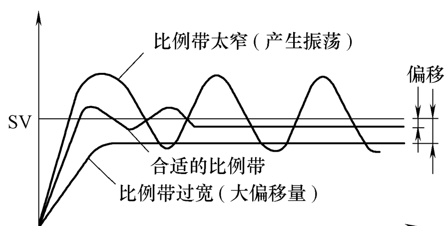


图 4-46　比例带对特性影响

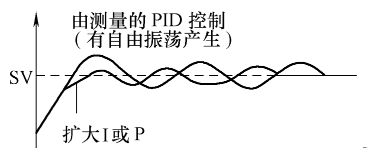


图 4-47　积分作用对特性影响

4. 微分作用

微分作用是增加系统的稳定。但如果产生了短周期振荡，可能是由控制系统响应快和微分作用过强造成的，此时调低微分作用，如图 4-48 所示。

此外，还有前馈滤波系数，输入输出位数等，也要做相应的设定。

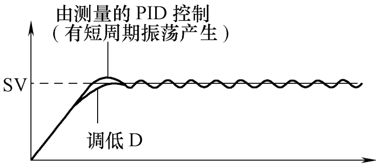


图 4-48 微分作用对特性影响

4.6.4 PID 指令执行

此指令不能放在 IL（02）和 ILC（03）、JMP（04）和 JME（05）指令之间，或放在子程序中，或放在步指令（STEP（08）/SNXT（09））中。否则不能执行。

当指令的执行条件 OFF 时，本指令不执行，但设定的参数保留，控制值由输出字 D 的内容确定。这时，直接写 D，改变它的值，可实现手动控制。

当指令的执行条件第一次从 OFF 到 ON 时，先读参数，初始化工作区，然后执行 PID 运算，并把结果值送 D。要注意的是，对 CS1 机之前的机型，一旦指令开始执行，参数改变将不起作用。

当指令的执行条件继续 ON 时，执行本指令，但只在设定采样时间到的周期才执行。图 4-49 示出 PLC 扫描周期与采样周期（设为 100ms）以及执行本指令间的关系。

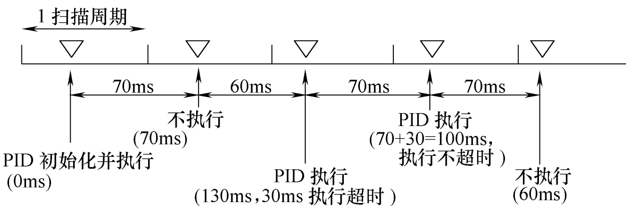


图 4-49 扫描周期与采样周期关系

如果 C 数据超出范围，将产生一个错误，PLC 错误标志位 ON。如果实际采样周期大于指定采样周期的两倍，将产生一个错误，错误标志置 ON，而 PID 控制继续执行。

PID 执行时进位标志置 ON。

对 CS1 机，执行 PID 动作后，操作变量超过上限，大于标志置 ON，此时计算结果以上限值输出。执行 PID 动作后，操作变量低于下限，小于标志置 ON，此时计算结果按下限值输出。

表 4-3 为执行 PID 指令时，在不同情况下，各标志位的取值。一般可通过对这些值的判断，得知本指令是否得以正确执行。

PID 指令使用实例 1：其梯形图程序如图 4-50 所示，它用于某电动机转速 PID 控制。PLC 为 CPM2ACPU60。

从图知，当“PID 指令控制”触点 ON，则执行本控制程序。这里有用得就只有一条 PID 指令。这 PID 指令有 3 个操作数：DM942，存实际值，十六进制格式；DM900，存控制参数，占 33 个字；DM950，存 PID 运算结果，十六进制格式。

表 4-3　执行 PID 指令时标志位状态

名　称	标　记	操　作
错误标志	ER	C 数据超出范围时置 ON 实际采样周期大于设定采样周期的两倍时置 ON 其它情况置 OFF
大于标志	>	执行 PID 动作后操作变量超出上限时置 ON 其它情况置 OFF
小于标志	<	执行 PID 动作后操作变量低于下限时置 ON 其它情况置 OFF
进位标志	CY	PID 控制正在执行时置 ON 其它情况置 OFF

这里的关键是设定好控制参数。本例的有关设定值为

DM 0900：#5F，控制的目标值，十六进制格式，不过这里的 5F 只是转速象征值；

DM 0901：#40，比例带，BCD 格式；

DM 0902：#100，积分常数，BCD 格式；

DM 0903：#80，微分常数，BCD 格式；

DM 0904：#10，采样周期，BCD 格式，单位 100ms，本例采样周期为 1s；

DM 0905：#0，反向控制；

DM 0906：#0，输入、输出范围为 8 位，即 0～FF。

图 4-50　PID 指令使用实例 1

这里还对 P_CY 及 P_ER 是否 ON 作了记录，目的是看参数设定是否正确。如图所示，如出现 LR1.00 ON，说明本指令已执行。如 LR1.03 ON，说明执行本指令出错，将得不到预期的结果。

可知，用 PID 指令实现控制，编程非常简单，一条指令加参数正确设定即可。

提示 1：如果控制参数设置得当，执行条件 ON，本指令执行，进位位 P_CY ON。可用它观测本指令是否已执行。也可人为地改变指令控制的输出值，即指令操作数 D 的内容，如不可改变，也说明指令已执行。

提示 2：对如 CPM2A 这样的 PLC，PID 指令为扩展指令，使用前应先对其功能号进行设定。 设定后还要下载给 PLC。 对 CPM2A 机，在下载前还应把 DM6602 的高字节设为 01，这意味着允许下载扩展指令功能码的设定。

图 4-51　PID 指令使用实例 2

PID 指令使用实例 2：其梯形图程序如图 4-51 所示。

从图知，在 000000 的上升沿（从 OFF～ON），工作区 D00209～D00238 按 D00200～D00208 中的参数（如下所示）进行了初始化。工作区被初始化后，将执行 PID 运算，并将操作变量输出到 0020。当 000000 为 ON 时，PID 控制依据 D00200～D00208 中的设置，以采样周期间隔执行，操作变量输出到 0020。但是，在 000000 置为 ON 后，如果改变比例带（P）、积分常量（TiK）、微分常量，用在 PID 计算中的 PID 常量将不会改变。

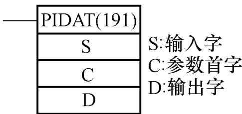
C：D00200　　设定值：300
C+1：D00201　　比例带：10.0%
C+2：D00202　　积分时间：120.0s
C+3：D00203　　微分时间：40.0s
C+4：D00204　　采样周期：0.5s
C+5：D00205　　反向操作（位00：0）/PID常数刷新时间=输入条件为ON
C+6：D00206　　（位01：0）/设定值=操作变量输出50%（位03：1）/2-PID
C+7：D00207　　参数=0.65（位4～位15：000H）
C+8：D00208　　操作变量输出范围：12位（位00～03：4H）
　　积分/微分常量设定时间（位04～位07：9H）
C+9：D00209　　输入范围：12位（第08～11位：4H）
～　　操作变量限位操作：无（第12位：0H）
C+38：D00238

提示：当000000为OFF时，其输出值，可用手工设定，如同手工对其操作。

4.6.5 PIDAT指令及其应用

1. PIDAT指令的格式及控制字

PIDAT是OMRON公司新型PLC新增的、参数可自整定的PID指令。其格式为



它的功能码为191。但目前该指令仅由CS1-H、CJ1-H、CJ1M和CS1D CPU支持。

它的控制参数共40个字，也应分布在同一连续的数据区中。表4-4示出其主要字的含义，及其选定。

表4-4　控制字含义

控制数据	项目	内容	设定范围
C	设定值（SV）	控制过程的目标值	二进制数据（与指定输入范围位数相同）
C+1	比例带	P作用参数 表示比例控制范围/总控制	0001～270FH（1～9999）： （0.1%～999.9%，以0.1%为单位）
C+2	Tik	表示积分作用强度的一个常量，数值增加，积分强度减小	0001～1FFFH （1～8191）（9999=积分操作不执行）①
C+3	Tdk	表示微分作用强度的一个常量，数值增加，微分强度减小	0001～1FFFH（1～8191）： （0000=不执行微分操作）
C+4	采样周期	设定执行PID运行的周期	0001～270FH（1～9999）： （0.01～99.99s，以10ms为单位）
C+5中位04～15	2-PID参数（α）	输入滤波系数，通常为0.65（即000设定），滤波效率随系数向0靠近而递减	000H：α=0.65 设置从100～163H表示最右边的两位的值被设定为从α=0.00到α=0.99②

（续）

控制数据	项　目	内　容	设定范围
C+5 中位 03	控制变量输出设定	设定 PV 等于 SV 时的控制变量输出	0:输出 0% 1:输出 50%
C+5 中位 01	PID 常量变化设定	PID 运算的比例带(P),积分常数(Tik)与微分常数(Tdk)允许变化的时刻	0:PID 指令开始执行 1:PID 指令开始执行和每一个采样周期
C+5 中位 00	PID 前向/逆向设定	决定比例作用的方向	0:正向　1:反向
C+6 中位 12	操作变量输出限制控制	决定操作变量输出是否应用限制操作	0:无效(无限制控制) 1:有效(有限制控制)
C+6 中位 08~11	输入范围	输入数据位数	0:8 位　3:11 位　6:14 位 1:9 位　4:12 位　7:15 位 2:10 位　5:13 位　8:16 位
C+6 中位 04~07	输出范围	输出数据位数(输出位数自动等于输入位数)	
C+6 中位 00~03	积分和微分单位	决定积分和微分常量的单位	1:采样周期倍数 9:时间(100ms)
C+7	输出变量下限	操作变量输出限制有效时的下限值	0000~FFFF(二进制)③
C+8	输出变量上限	操作变量输出限制有效时的上限值	0000~FFFF(二进制)③
C+9 位 15	AT 命令位	控制位开始自动调整 ·设置 AT 命令位为 1,执行自动调整。(当 PIDAT(191)执行时,自动调整起动) ·自动调整完成,该位自动置 OFF 如果 AT 命令位手动置 OFF,自动调整中断。在这种情况下,当自动调整中断,如果参数已经计算出来,PID 常数将允许改变	作为控制位: 0→1 执行自动调整 1→0 中断自动调整 (当自动调整完成时,PID(191)自动将位置 OFF) 作为标志: 0:自动调整没有执行 1:自动调整正在执行
C+9 位 00~11	AT 计算增益	设置该参数是调整 PID 计算结果基值到存储值。通常将该参数设置成默认值(0000) ·强调稳定性,增加该值 ·强调响应,减少该值	0000H:1.00(默认值) 0001~03E8H(1~1000):(0.01~10.00,以 0.01 为单位)
C+10	限制周期迟滞	当限制周期产生时设置迟滞。对反向操作默认设置置 ON,MV 有 SV-20% 迟滞。因为 PV 不稳定,而不能产生正确的限制周期,增加该设置。然而如果限制周期迟滞高于常值,AT 的准确性将下降	0000H:0.20%(默认值) 0001~03E8H(0.01~10.00%,以 0.01 为单位) FFFH:0.00% 注:百分数与输入范围相对应

表注同表 4-2。

执行条件为 ON 时，PIDAT（191）按照 C 中设置的参数（设定值，PID 常量等）在两个自由度上对目标值执行 PID 控制，从输入字 S 的内容中得到指定输入范围的二进制数据，并根据设定参数执行 PID 运算。计算结果以控制变量的形式传给输出字（操作变量）D。

在执行条件由 OFF 变为 ON 时，读取设定参数，如果这些设定参数超出了许可范围，PLC 的错误标志位将置 ON。如果这些设定参数在许可范围内，PID 按初始值运算，此时未运行缓冲操作，该操作在执行 PID 过程后作用于控制变量（缓冲操作过程指为了避免突然变化造成相反效果，而渐近地、持续地改变操作变量的过程）。

当执行条件变 ON 时，计入设定采样周期的 PV 值并完成处理。如图 4-52 所示。

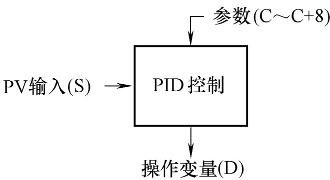


图 4-52 处理框图

2. 自动整定功能

在每一个周期都会检查自动命令位（C +9 的位 15）的状态。如果自动整定命令位 ON，PIDAT（191）中断用户设置常数的 PID 控制，开始自动整定 PID 参数（当自动整定时，SV 的变化将不会被自动反映出来）。

自动整定用到限制周期方法。PIDAT（191）强制使控制变量发生变化（最大值控制变量←→最小值控制变量），监视控制系统的特性。从观察到的特性中计算出 PID 常数，新的 P，I 和 D 参数自动存储到 C +1、C +2 和 C +3。此刻，自动命令位（C +9 的位 15）置为 OFF，在 C +1、C +2、C +3 中，用新的常数恢复 PID 控制。

图 4-53 所示的流程表为自动整定的过程。

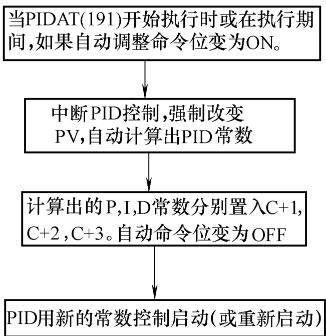


图 4-53 自动整定的过程流程表

注：1. 在自动调整期间，如果通过将 AT 命令位置 OFF 来中断自动调整，PID 控制将起动在自动调整之前的 PID 参数。

2. 如果发生自动执行错误，PID 控制将起动在开始调整前使用过的 PID 参数。

在上述注 1 和注 2 中，当中断自动调整时，PID 参数已经计算出来，它们将自动有效。

3. PID 控制

C +6 的第 8 ~11 位中设置的输入范围，指定了 16 位以内的 PV 输入（S）的有效输入数据位数。例如，如果指定输入范围为 12 位（十六进制 4），那么从 0000H ~0FFFH 的范围对 PV 是有效的。而大于 0FFFH 值时，将作为 0FFFH 处理。设定值的范围也依赖于输入范围的选定。

测量值（PV）和设定值（SV）是从 0000H 到输入范围的最大值的二进制无符号数。C +6 的第 0 ~3 位中设置的输出范围，指定了 16 位以内的操作变量有效输出数据位数。例如，如果指定输出范围为 12 位（十六进制 4），那么从 0000H ~ 0FFFH 的范围将被输出为控制变量。

对只是比例操作，当 PV 等于 SV 时，变量可被指定为如下：

0：输出 0%

1：输出 50%

比例操作方向可设定为正向或反向。

控制变量输出可以指定为上限制和下制限。

采样周期可以设定为以 10ms（0.01 ~ 99.99s）为单位的值，但实际的 PID 运行由采样周期和 PID（190）指令执行时间（每个周期）共同决定。

使 PID 常量变化的时刻可以设置成下列之一：

1）PID 指令执行开始。

2）PID 指令执行和每个采样周期开始。

在每个采样周期（即 PID 指令执行期间）仅比例带（P）、积分常量（Tik）、微分常量（Tdk）可以变化。时刻由 C + 5 的位 1 设置。

当手动改变 PID 常数时，设置 PID 常量使能位（C + 5 的位 1）为 1，这样 C + 1，C + 2，C + 3 的值在 PID 计算的每个采样期间可被刷新。这种设置在自动调整后也可以手动调整 PID 常数。

在 PID 参数（C ~ C + 38）中，当执行条件为 ON 时，下列参数可以被改变。当任何其它的值被改变后，确保变化条件从 OFF 到 ON，以便新的设置允许。

在 C 中设置值（SV）仅在 PID 控制可以被改变。在自动调整期间，SV 变化不会被反映出来。

执行条件像一个停止—运行信号一样控制着 PIDAT（191）的执行。在 C + 9 ~ C + 38 被初始化后，若下一个周期执行条件仍为 ON 则执行 PID 运算。因此，当使用常 ON 标志（A1）作为 PIDAT（191）的执行条件时，在操作开始时要有一个单独的 C + 9 和 C + 38 初始化过程。如果 C 数据超出范围，将产生一个错误，错误标志置 ON。

如果实际采样周期大于指定采样周期的两倍，将产生一个错误，错误标志置 ON，而 PID 控制继续执行。PID 控制被执行时进位标志置 ON。

执行 PID 运算后，控制变量超过上限，大于标志置 ON，此时计算结果以上限制值输出。如控制变量小于下限，小于标志置 ON，此时计算结果按下限值输出。

表 4-5 所示为执行 PIDAT 指令时，在不同情况下，各标志位的取值。一般可通过对这些值的判断，得知本指令是否得以正确执行。

表 4-5　执行 PIDAT 指令时标志位状态

名　称	标　记	操　作
错误标志	ER	C 数据超出范围时置 ON 实际采样周期大于设定采样周期的两倍时置 ON 其它情况置 OFF
大于标志	>	执行 PID 操作后,控制变量超出上限时置 ON 其它情况置 OFF
小于标志	<	执行 PID 操作后,控制变量小于下限时置 ON 其它情况置 OFF
进位标志	CY	PID 控制正在执行时置 ON 其它情况置 OFF

4. 应用实例

图 4-54 所示为 PIDAT 指令的应用实例。如图所示，在 CIO 000000（OFF→ON）上升沿，D00211 ~ D00240 工作区按照设置在 D00200 ~ D00208 的参数（如下所示）初始化，在

工作区已经初始化后，PID 控制执行，控制变量输出到 CIO 00200。

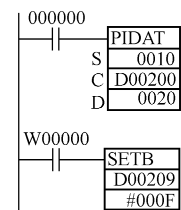


图 4-54　PIDAT 指令的应用实例

当 CIO 000000 置 ON，按照在 D00200 ~ D00210 的参数设置，在采样周期期间 PID 控制执行，将控制变量输出到 CIO00200。在 CIO00200 置 ON 后，即使改变比例带（P），积分常量（TiK），微分常量，用于 PID 运算的常数将不会改变。

在 W 000000（OFF→ON）上升沿，SETB（532）置 D00209（C +9）位 15 为 ON，开始自动调整。当自动调整完成，计算 P、I、D 常数写到 C +1、C +2、C +3 中。并置 D00209（C +9）位 15 为 OFF，之后，PID 将在新常数下，进行运算。

图 4-55 所示为有关参数设定。

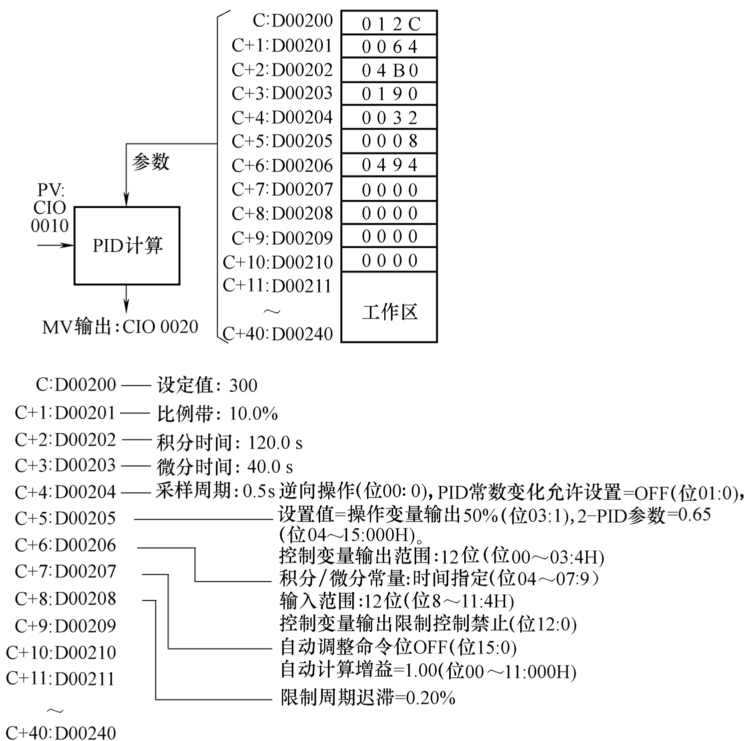


图 4-55　PIDAT 指令数据设定

图 4-56 所示为 PID 控制及 PID 参数自整定的实际过程。

值得注意的是，有时使用 PID AT 指令，PID 执行但 AT 功能不执行。这里问题在于，在 AT 位（C +9 的 15）ON，PV 在 SV 作上下波动 3 次后，AT 完成，即 AT 位为 OFF。如果 PV 没有上下波动，则 AT 不执行。

4.6.6　使用 PID 指令有关细节

使用 PID 指令进行 PID 控制，虽只使用一个指令及作有关设定。但也还有一些细节要考虑与处理：

1. 数据格式转换

尽管 PID 指令输入、输出用的格式为十六进制码，与模拟量输入、输出单元的数据格式是相同的，为使用 PID 指令提供了方便。但是，还是存在一些不那么对应的情况。如用脉冲作输入信号，其格式用的就不是十六进制码，而是 BCD 码。调宽脉冲输出，其格式用的也不是十六进制码，而是 BCD 码。而 PID 指令的输入（源字）、输出（结果字）都是十六进制码。这时，在使用 PID 前，要做一些数据格式转换，如必要也可能还要作比例换算，以便于执行 PID 指令时，使用有关数据。同时，在执行 PID 指令得出结果后，也要把这个结果值进行换算，以便于为有关输出使用。

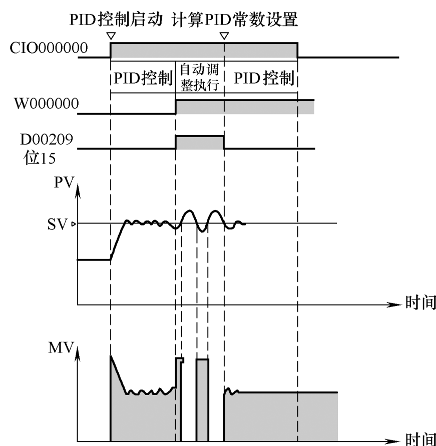


图 4-56　PID 参数自整定实际过程

把 BCD 码转换为十六进制码，可用 BIN 指令。反之，可用 BCD 指令。图 4-57 所示为有关转换指令的使用情况。图 4-57a 为执行 PID 指令前的转换，它把输入数据 BCD 码 DM666 转换为十六进制码数据 DM716。用它作 PID 指令的原数据。图 4-57b 为执行 PID 指令后的转换，它把 PID 指令执行后的结果数据十六进制码 DM718，转换为 BCD 码数据 DM500，以作为实际控制输出。

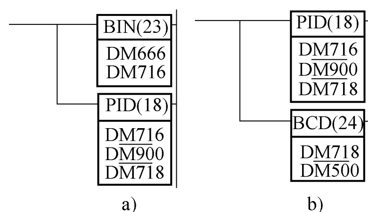
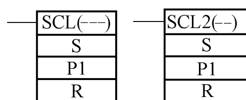


图 4-57　数据转换梯形图程序

PID 指令执行后的转换，也可用 SCL 及 SCL2 两个指令。这两个指令的格式为



这里的 S 为原数据，R 为结果数据，P1 ~ P3 为控制字。此两个指令的功能是：根据 P1 ~ P2 制定线性关系，把不带（SCL）或带（SCL2）符号的十六进制数码数据（S）转换为 BCD 码数据（R）。图 4-58 显示了其间（用 SCL）的关系。

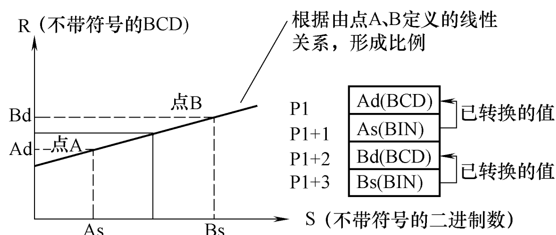
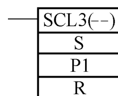


图 4-58　用 SCL 指令实现转换后两数对应关系

PID 指令执行前的转换，也还可用 SCL3 指令。这个指令的格式为



这里的 S 为原数据，R 为结果数据，P1 ~ P5 为控制字。这个指令的功能是：根据指定线性函数，将有符号 BCD 码数据（S）转化为有符号十六进制码数据（R）。

控制字的含义及其取值为

P1：为偏移值，有符号十六进制数，取值范围为：8000 ~ 7FFF；

P2：为 ΔX，BCD 码，取值范围为：0001 ~ 9999；

P3：为 ΔY，十六进制码，取值范围为：8000 ~ 7FFF；

P4：为最小转换，十六进制码，取值范围为：8000 ~ 7FFF；

P5：为最大转换，十六进制码，取值范围为：8000 ~ 7FFF。

其转换公式为

$$R = \frac{\Delta Y}{\Delta X\text{ 的 BCD 码}} \times (S\text{ 的二进制转换码}) + (\text{偏移})$$

图 4-59 显示了在正、0 及负偏移情况下，其间的关系。

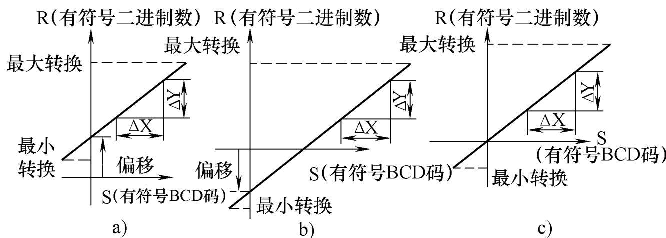


图 4-59 不同的对应关系

a）正偏移 b）负偏移 c）0 偏移

CJ1 机还有静带控制指令 BAND（681）及静域控制指令 ZONE（682）。可分别用处理输入、输出死区问题。有关细节，可参阅有关说明书。

2. 参数变换

执行 PID 指令，当偏差值不同时，所选的参数最好也有所不同。如比例带、积分常数等，在偏差大时，可选小一些，以强化它的作用。而偏差小时，或控制输出已将接近开环测定的数值，这些参数可与偏差大些，以弱化它的作用。图 4-60 所示即为此类程序。如图所示，当“偏差值 1”大于 4 时，把 250 赋给 DM901，500 赋给 DM902；反之，把 20 赋给 DM901，50 赋给 DM902。这里的 DM901、DM902 分别为“比例带”、“积分常数”，正是 PID 控制参数。

但是，CS1 机以前的 PLC，一旦指令开始执行，参数改变将不起作用。为此，要对 PID 指令作再执行处理，即当参数改变时，先使执行 PID 指令的条件 OFF，然后再令其 ON。图 4-61 即为此类程序的一个。

从图知，当 DM901、DM902 的值改变时，LR1.01、LR1.02 将 OFF 一个扫描周期，用此即可使新参数起作用。

3. 输出值控制

尽管作了数据变换或转换，但有时，还是难以做到控制输出值与所驱动的对象完全适应。如有这样的情况，模拟量输出单元的输出范围为 0 ~ 10V，而对象所用的控制输入只能

为 0～5V。这时就必须对输出值进行控制，如图 4-62 所示程序，当“输出值”大过 50 时，则用 50 传送给它，确保“输出值”不超过 50。其程序如图 4-62 所示。

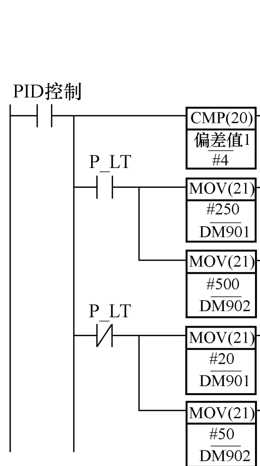


图 4-60　参数变换梯形图程序

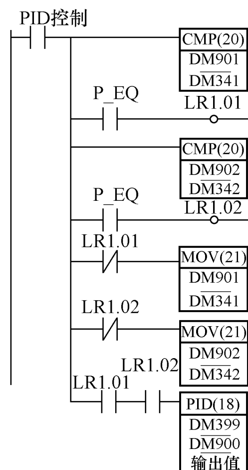


图 4-61　使新参数立即起作用梯形图程序

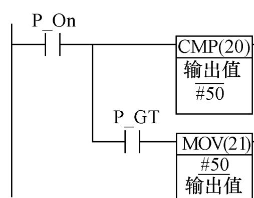


图 4-62　输出值控制梯形图程序

CJ1 机还有限幅指令 LMT（680），也可用以作此类问题的处理。该指令的含义及其使用请参阅有关说明书。

4. 手动、自动无扰动切换

实际系统除了自动控制，有时还要求手动控制。而且，这两者切换时，最好系统的控制输出不要突变，以免系统受到不应有的冲击。参数变化时也有这个问题。

怎么做到这一点呢？一般讲在两种情况下转换，冲击将少一些。即：在偏差值为 0，或尽量小时转换；或控制输出值相等或接近相等时转换。

为此，在手动向自动转换时，可观测自动的给定值与实际值相差大否？太大时，应使其调到相接近时，切换。自动到手动转换，应使手动的输出初值设为当时自动的控制输出值。

5. 多种控制算法配合使用

PID 控制虽然有很多优点。但由于系统的非线性等诸多原因，完全用 PID 控制有时难以满足实际要求。所以，也可考虑依不同的偏差值，或依不同的工况，有时用 PID 控制，有时也可用别的控制。多种控制配合使用，以获得更好的控制效果。

4.7　模拟量 PID 硬件单元

关键词：PID 单元、温度控制单元、回路控制单元、内插板、功能软件模块

以上介绍的 PID 控制都是用 PLC 的程序实现的。随着 PLC 用于模拟量控制的增加，多数 PLC 厂家都开发有专用于模拟量控制的模块，如 PID 模块（单元）、温度控制模块等。这些模块有自己的 CPU，自己的内存。可用自身的 CPU 实施 PID 控制。近期，OMRON 公司还开发有回路控制单元及回路控制内插板，可快速地实现几十路的 PID 控制。

4.7.1　PID 单元

一般大、中型 PLC 多有此类模块。以下 OMRON 200 系列机用的 C200H-PID 单元为例，对此类单元作简要介绍：

它有 3 种型号 001、002、003。分别为半导体输出、电压输出及电流输出。输入为标准信号，可选定，有：4 ~ 20mA、1 ~ 5V 、0 ~ 5V、0 ~ 10V 等。图 4-63 所示为该模块的外观图。可知，它与其它 C200H 特殊单元的外观是类似的。

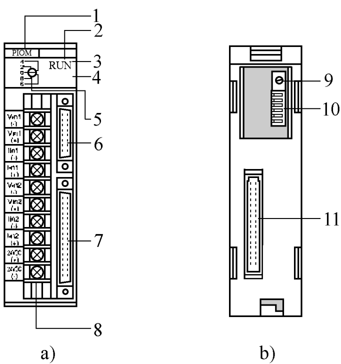


图 4-63　PID 单元

a）前面板　b）后面板

1—模块标志　2—盖板　3—运行指示灯　4—SW2 放置在盖板之下用以切换内存及设置控制方向　5—SW1 设置单元号开关　6—数据设定器连接器　7—输出连接器　8—输入端子　9—SW202 输入类型选择开关　10—SW203 操作与功能选择开关　11—后连接器

这里的 SW1 用以设定单元号。单元号只能在 0 ~ 9 之间选定，而且不能与别的单元号重复。

这里的 SW2 有两个开关，SW2-1 及 SW2-2。SW2-1 用以设定内存。其 ON 时，其内存可用命令改变，OFF 时，其内存预定值不变。SW2-2 用以设定方向。其 ON 时，PLC 设定。其 OFF 时，数据设定器设定。

这里的 SW202 用以输入信号类型设定。见表 4-6。

它有两个回路，可分别独立工作。但不能用作串级控制，这是它的不足。有的厂家的 PID 模块则可这么做。也可只用回路 1，而把回路 2 设为“Disabled“，不用。但不能只用回路 2，而不用回路 1。

表 4-6　输入信号类型设定

开关号	回路 1	回路 2	开关号	回路 1	回路 2
0	4 ~ 20mA	4 ~ 20mA	5	4 ~ 20mA	0 ~ 5V
1	1 ~ 5V	1 ~ 5V	6	4 ~ 20mA	0 ~ 10V
2	0 ~ 5V	0 ~ 5V	7	1 ~ 5V	0 ~ 5V
3	0 ~ 10V	0 ~ 10V	8	1 ~ 5V	0 ~ 10V
4	4 ~ 20mA	1 ~ 5V	9	0 ~ 5V	0 ~ 10V

该模块还可配置数据设定器。也可不配置。此设定器的外观很像温控表，如图 4-64 所示。

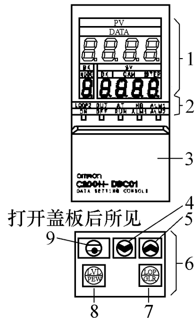


图 4-64　数据设定器

1—数据显示　2—操作指示灯　3—盖板　4—DOWN 键　5—UP 键　6—操作键　7—LOOP 键　8—LEVEL 键　9—DISPLAY 键

从图知，此设定器有显示屏，指示灯及操作按钮等。当 PID 控制模块设置为用设定器操作时，可用它实施对 PID 控制模块的操作。其显示屏不但可显示当前值及设定值，还可用作设定的操作的信息提示。

该模块有自己的 CPU，可实现 PID 控制。同时，可实现 PID 参数自整定。而且在实现控制时，也具有 OMRON 公司 PID 前馈控制特性。也还可设定成 ON/OFF 控制。

该模块有自己的内存，有 EERAM 及 RAM，前者可保持数据，但写的次数受限制。所以，应先用 RAM 操作。

提示：2002 年，在赤峰电厂使用此模块时，曾因操作不当，对 EERAM 改写的次数太多，而相继报废 3 个模块。 以前 C200H 的 EERAM 也有写次数限制。 所以，此问题不能小看。

该模块可实现高速采样，两个回路同时工作时，采样周期可小到 100ms。

该模块它自带有数字滤波，可使其输入信号少受干扰。

该模块设定值最多可设为 8 段。这意味着可用其进行程序控制，实现被控对象按程序变化。对应的还可设定相应的报警值。

该模块工作时，PLC 则可通过程序与其交换数据。

等等。

该模块的使用的 IR 地址与单元号的设定有关。具体关系如图 4-65 所示。

图 4-65　IR 地址与单元号关系

当 SW2-1 OFF 时，10 个 IR 字的功能见表 4-7。

表 4-7　当 SW2-1 OFF 时，10 个 IR 字的功能　（n = 100 + 10 × 单元号）

<table>
<tr><td rowspan="2">I/O字</td><td rowspan="2">Word</td><td colspan="16">Bit</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>09</td><td>08</td><td>07</td><td>06</td><td>05</td><td>04</td><td>03</td><td>02</td><td>01</td><td>00</td></tr>
<tr><td rowspan="6">输出</td><td rowspan="2">n</td><td colspan="16">Loop 1 SP(see note)</td></tr>
<tr><td colspan="4">0 ~ 9, F</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td></tr>
<tr><td rowspan="2">n + 1</td><td colspan="16">Loop 2 SP(see note)</td></tr>
<tr><td colspan="4">0 ~ 9, F</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td></tr>
<tr><td rowspan="2">n + 2</td><td colspan="4" rowspan="2">回路 1 执行段号</td><td colspan="4" rowspan="2">回路 2 执行段号</td><td colspan="2">回路 1</td><td colspan="2">回路 2</td><td colspan="4" rowspan="2">0</td></tr>
<tr><td>MAN</td><td>RUN</td><td>MAN</td><td>RUN</td></tr>
<tr><td rowspan="10">输入</td><td rowspan="2">n + 3</td><td colspan="16">Loop 1 PV(see note)</td></tr>
<tr><td colspan="4">0 ~ 9, F</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td></tr>
<tr><td rowspan="2">n + 4</td><td colspan="16">Loop 2 PV(see note)</td></tr>
<tr><td colspan="4">0 ~ 9, F</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td></tr>
<tr><td rowspan="2">n + 5</td><td colspan="16">Loop 1 SP(see note)</td></tr>
<tr><td colspan="4">0 ~ 9, F</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td></tr>
<tr><td rowspan="2">n + 6</td><td colspan="16">Loop 2 SP(see note)</td></tr>
<tr><td colspan="4">0 ~ 9, F</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td><td colspan="4">0 ~ 9</td></tr>
<tr><td rowspan="2">n + 7</td><td colspan="4" rowspan="2">回路 1 执行段号</td><td colspan="4" rowspan="2">回路 2 执行段号</td><td colspan="4" rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td colspan="2">SW2</td></tr>
<tr><td>2</td><td>1</td></tr>
</table>

（续）

I/O字	Word	Bit															
		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
输入	n+8	回路1状态															
		0	传感器错	0	0	0		MAN	RUN	0	0	0	控制输出	AT	0	AL1	AL2
	n+9	回路2状态															
		0	传感器错	0	0	0		MAN	RUN	0	0	0	控制输出	AT	0	AL1	AL2

注：4位BCD码，小数点位由“小数点位”指定，最高位为F，为负值。

当SW2-1ON时，10个IR字的功能见表4-8。

表4-8　当SW2-1ON时，10个IR字的功能　（n=100+10×单元号）

I/O字	Word	Bit															
		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
输出	n	读写指令															
		读写		回路号		段号				指令							
	n+1	写数据(see note)															
		0~9,F				0~9				0~9				0~9			
	n+2	回路1执行段号				回路2执行段号				回路1		回路2		0		0	写要求
										MAN	RUN	MAN	RUN				
输入	n+3	回路1现值(see note)															
		0~9,F				0~9				0~9				0~9			
	n+4	回路2现值(see note)															
		0~9,F				0~9				0~9				0~9			
	n+5	读数据(see note)															
		0~9,F				0~9				0~9				0~9			
	n+6	0				写错	0	0	写完成	0				0	0	0	读完成
	n+7	回路1执行段号				回路2执行段号				0				0	0	SW2	
																2	1
	n+8	回路1状态															
		0	传感器错	0	0	0		MAN	RUN	0	0	0	控制输出	AT	0	AL1	AL2
	n+9	回路2状态															
		0	传感器错	0	0	0		MAN	RUN	0	0	0	控制输出	AT	0	AL1	AL2

注：4位BCD码，小数点位由“小数点位”指定，最高位为F，为负值。

PLC 可使用这些地址向模块发送命令（指令）。命令由命令码及命令参数组成。有关命令码的功能及其参数选用见表 4-9。

表 4-9　命令码及参数选用

项　目	命令码		写	读	回路号	段号	实际数范围	设定范围	默认值
设定值	0	0	Yes	Yes	Yes	Yes	从标定低限到标定高限		OEU
手动输出值	0	1	Yes	Yes	Yes	No	－5%～105%	F005～0105	0%
指定段号	0	2	Yes	Yes	Yes	No	1～8	0001～0008	1
报警值设定 1	0	3	Yes	Yes	Yes	Yes	Alarm mode 1,4,5:0～9999 Other:－999～9999	0000～9999 F999～9999	OEU
报警值设定 2	0	4	Yes	Yes	Yes	Yes			OEU
输入偏移	0	5	Yes	Yes	Yes	Yes	－999～9999	F999～9999	OEU
比例带	0	6	Yes	Yes	Yes	Yes	0.0～999.9	0000～9999	10.0% FS
积分时间	0	7	Yes	Yes	Yes	Yes	0～9999s		240s
微分时间	0	8	Yes	Yes	Yes	Yes	0～9999s		40s
小数点位置	1	2	Yes	Yes	Yes	No	0～3	0000～0003	1
标定低限	1	3	Yes	Yes	Yes	No	－999～(扫描上限值－1 数位)		0
标定高限	1	4	Yes	Yes	Yes	No	(扫描下限值＋1 数位)～9999		1.000
控制周期	1	7	Yes	Yes	Yes	No	1～99s	0001～0099	20s
滞后	1	8	Yes	Yes	Yes	Yes	0.0～100.0	0000～1000	0.2% FS
报警滞后	1	9	Yes	Yes	Yes	No	0.0～100.0	0000～1000	0.1% FS
数字滤波	1	E	Yes	Yes	Yes	No	0～100	0000～0100	0s
控制输出变量监控	2	0	No	Yes	Yes	No	0.0～100.0%	0000～1000	0.0%
自整定开始/停止	2	1	Yes	No	Yes	No	AT 开始＝0001 AT 停止＝0000		—
段复制	2	2	Yes	No	Yes	No	—		—
输入类型监控	2	3	No	Yes	No	No	0～9	0000～0009	针对 SW202 设定
报警方式 1	2	4	Yes	Yes	Yes	No	0～9	0000～0009	2
报警方式 2	2	5							

注：Yes：可以；No：不可以或不要求。

为了正确使用该模块，要按以下步骤进行操作：

（1）按要求，在其前后板的设定开关上，作好相应的设定。SW2-2 处 OFF 位置时，数据设定器才可对其进行操作。

（2）关断电源，把模块安装到 PLC 的底板上。

（3）按规定接好输入线。24V 的电源必须连接，否则设定器无法工作。

（4）安装数据设定器（如果使用它）。

（5）加 PLC 电源及 24V 电源。置 PLC 于编程模式。

（6）用数据设定器作必须改变的参数设定。

（7）连接输出线，接通电源，用编程器或相应设备置回路运行位（RUN BIT）ON，开始测试与调整，直到达到要求。这时，段号置执行段号，并不要用设定器改变它。要改，可用编程器或用 PLC 程序。

（8）根据内存分配，建立用户程序，以可进行数据设定、监控、信号转换等，并置 SW2-2 ON，把对模块的操作权交给 PLC。

（9）开始运行。

可知，使用 PID 模块实现 PID 控制。主要的工作是设定、数据传送等。要编的 PLC 程序主要是运用命令对模块作些操作，工作量是不大的。

4.7.2 温度控制单元

温度量是很常用的模拟量，在工业控制中，常要对其进行控制。在模拟量 PID 控制硬件单元中，各 PLC 厂家多开发有专用于温度控制的硬件单元，即这里将介绍的温度控制单元。它不仅直接读入温度信号，而且可依据预置的控制方式与要求的温度值，进行温度控制。

CJ1W-TC 系列模块就是这样的温度控制单元的一种。是 OMRON 公司 CJ1 机的一种特殊 I/O 单元。它直接接受来自热电偶或铂电阻温度计的输入，执行相应的 PID 控制，并通过开路集电极送出输出结果。

图 4-66 所示为 CJ1W-TC 系列模块的正面视图（图 4-66a）与侧面视图（图4-66b）。

由于 CJ1 机为无底板的模块式 PLC，所以，它的模块间连接器在侧面。从正面视图可看出，它的面板上有单元号、DIP 及输入类型开关。还有 3 个状态指示灯及若干输出指示灯。还有接线端子。状态指示灯名称及其含义见表 4-10。

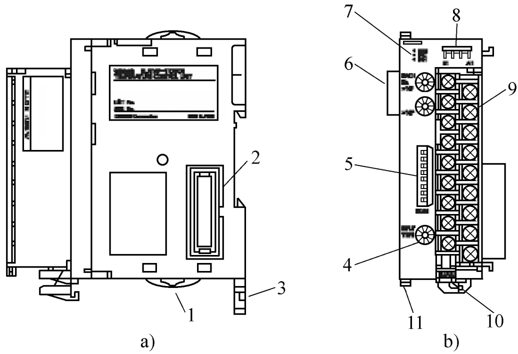


图 4-66 CJ1W-TC 系列模块

1、11—滑动锁扣 2—连接器 3—DIN 安装栓 4—输入类型开关 5—DIP 开关 6—单元号开关 7—状态指示灯 8—输出指示灯 9—端子板 10—端子板锁定杆

表 4-10 状态指示灯含义

指示灯	名　　称	颜色	状态	含　　义
RUN	运行指示器	绿	亮	正常操作状态
			不亮	温度控制停止
ERC	温度控制单元出错	红	亮	温度控制单元本身出现差错,如传感器出错或初始化出错
			不亮	正常操作状态
ERH	CPU 出错	红	亮	CPU 单元中出现错误
			不亮	正常操作状态

1. 类型

CJ1W-TC 系列模块有两种主要单元类型：一种提供 4 个控制回路，另一种提供 2 个带加热器断线检测功能的控制回路。两种类型都有和热电偶（R，S，K，J，T，B 或 L）相配的型号及和铂电阻温度计（JPt100 或 Pt100）相配的型号，都可使用 NPN 输出和 PNP 输出。也可进行 PID 控制参数的自动整定。

图 4-67 表示 CJ1W-TC001 温度控制单元（4 个控制回路，热电偶输入，NPN 输出）和 CJ1W-TC103 温度控制单元（2 个控制回路，带加热器烧断检测，铂电阻温度计，NPN 输出）的基本系统。

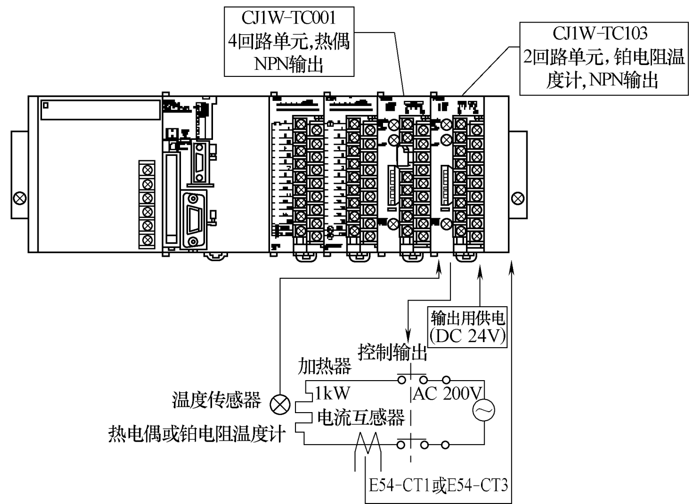


图 4-67　温度控制单元控制系统

要注意的是：第一、必须使用一个 OMRON E54-CT1 或 E54-CT3 电流互感器，不要使用任何其它电流互感器。第二、将回路的停止位转为 ON 以停止温度控制。如果正在使用 PID 控制，而且用一个操作开关输入到加热器使加热器转为 OFF，则 PID 控制性能是反作用的。

2. 规格

CJ1W-TC 系列模块共同规格见表 4-11。不同类型的不同规格见表 4-12。

表 4-11　CJ1W-TC 系列模块共同规格

<table>
<tr><th>项　目</th><th colspan="4">规　格</th></tr>
<tr><td>最多单元数</td><td colspan="4">10 单元/机架 max.（CPU 机架或扩展机梁）</td></tr>
<tr><td rowspan="4">用于数据储存/交换的 CPU 单元数据区</td><td rowspan="2">特殊 I/O 单元区（960 字）
CIO 2000 ~ CIO 2959</td><td rowspan="2">20 字/单元用于固定的数据交换（6 个输出字和 14 个输入字）</td><td>CPU 单元到温度控制单元</td><td>· 设定点（SP）· 运行/停止控制
· 操作指令 · 起动/停止 AT
· 写指令
· 加热器烧断电流设定</td></tr>
<tr><td>温度控制单元到 CPU 单元</td><td>· 过程值（PV）· 状态
· 设定点（SP）· 加热器电流检测</td></tr>
<tr><td rowspan="2">分配到特殊 I/O 单元的 DM 字（9600 字）
D20000 ~ D29599</td><td>当电源转为 ON 或单元重新起动时传送 10 个字/单元</td><td>CPU 单元到温度控制单元</td><td>· 报警模式
· 报警滞后</td></tr>
<tr><td>90 字/单元用于常规的数据交换</td><td>CPU 单元和温度控制单元二路传送</td><td>· 报警值　· 比例带
· 输入补偿值　· 积分时间
· 控制周期　· 微分时间
· 灵敏度　· 输出检测</td></tr>
</table>

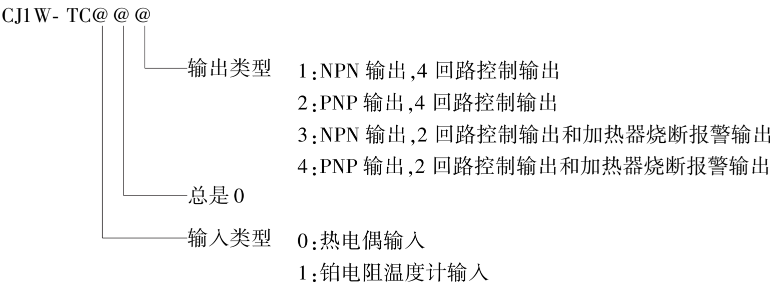
（续）

项　　目	规　　格	
控制输出和加热器烧断报警输出	NPN 或 PNP 输出，都具有短路保护（见注 1）　漏电流：0.3mA max 外部供电电压：24VDC +10%/-15%　剩余电压：3V max 最大开关容量，100mA（每个输出）	
温度控制方式	ON/OFF 控制或带二个自由度的 PID 控制（用单元的 DIP 开关的第 6 针设定）	
控制操作	正向或反向操作（用单元的 DIP 开关第 4，5 针设定）	
运行/停止控制	支持（由 CPU 单元通过在特殊 I/O 单元区的分配位来控制）	
在编程模式中的 CPU 单元操作	温度控制单元可设定为当 CPU 单元处于编程模式时继续工作或停止操作（用单元的 DIP 开关的第 1 针设定）	
用于操作输出的自动/手动开关	无	
PID 常数的自动调整（AT）	支持（由 CPU 单元通过特殊 I/O 单元区的分配位来控制）	
灵敏度（当使用 ON/OFF 控制时）	0.0～999.9℃或℉（0.1℃或℉单位）	
比例带	0.1～999.9℃或℉（0.1℃或℉单位）	
积分（重定）时间	0～9999s（1s 为单位）	
微分（变化率）时间	0～9999s（1s 为单位）	
控制周期	1～99s（1s 为单位）	
采样周期	500ms（4 回路）	
输出刷新周期	500ms（4 回路）	
显示刷新周期	500ms（4 回路）	
输入补偿值	-99.9～999.9℃或℉（0.1℃或℉为单位）	
报警输出设定范围	-999～9999℃或℉（1℃或℉为单位） 当使用铂电阻温度计或使用 K 或 J 型热电偶以小数点形式显示时，设定范围为 -99.9～999.9℃或℉（以 0.1℃或℉为单位）	
对 CPU 单元循环时间的影响	0.4ms	
型号	CJ1W-TC00@	CJ1W-TC10@
温度传感器	热电偶：类型 R，S，K，J，T，L 和 B	铂电阻温度计：类型 Pt100 和 JPt100

（续）

项　　目	规　　格	
指示精度	摄氏：±0.3% PV 或 ±1℃（大者）±1 位 max 华氏：±0.3% PV 或 ±2℉（大者）±1 位 max ·当使用 L 型热电偶或低于 -100℃ 下使用 K 或 T 型热电偶时精度为 ±2℃　±1 位 max ·使用 R 或 S 型热电偶在 200℃ 以下时，精度为 ±3℃ ±1 位 max ·低于 400℃ 时 B 型热电偶会是不准确的@	摄氏：±0.3% PV 或 ±0.8℃（大者）±1 位 max 华氏：±0.3% PV 或 ±1.6℉（大者）±1 位 max

注：1. 型号的最后 3 位标明单元的特点。



2. 热电偶的指示精度

·给出的精度等级是指温度控制单元和冷端补偿器（在端子板上）一体使用时的。必须将单元和端子板一起使用，单元和端子板上附有系列号以帮助保持一体性。

·当热电偶型温度控制单元返回检修时，必须将单元和端子板（带冷端补偿器）作为一体返回。

此外，有关加热器烧断（HB）报警的有关规格见表 4-12。

表 4-12　加热器烧断（HB）报警的有关规格

项　　目	规　　格
加热器最大电流	单相 AC,50A
输入电流指示精度	满量程的 ±5% ±1 位 max
加热器烧断报警设定范围	0.1 ~49.9A(单位 0.1A) 如果设定值为 0.0A 或 50.0A,加热器烧断检测功能将不起作用 (当 SV 值为 0.0A 时加热器烧断报警将为 OFF,当 SV 为 50.0A 时加热器烧断报警将为 ON)
可检测的最小 ON 时间(见注)	200ms

注：如果控制输出 ON 的时间小于 200ms，加热器的烧断检测功能将不起作用，而且加热器电流测量也将不执行。

3. 特点

（1）可直接连接温度传感器：可以将温度传感器直接连接到温度控制单元（2 或 4 个输入）。有两种

型号用于热电偶（R、S、K、J、T、B 和 L 热电偶），两种型号用于铂电阻温度计。

（2）控制方式可选定：可进行 PID 控制，也可进行 ON/OFF 控制。这可用单元前面板上的一个开关（DIP 开关的第 6 针）选定。

如用 ON/OFF 控制，首先要设定一个设定点。在反向操作时，达到设定值时，温度控制器将控制输出转为 OFF，控制输出关断时系统温度将开始降落。当系统温度降到低于设定点时，控制输出将重新转为 ON，这种 ON/OFF 操作是围绕设定点反复动作的。

为了避免这种反复动作过于频繁，可设定控制灵敏度。它决定了在控制输出重又转为 ON 之前，系统温度降落到低于设定点多远（见图 4-68）。

如用 PID 控制，其采样周期可设定，默认值为 500ms ，与 CPU 单元的循环时间无关。并对 CPU 单元的循环时间没有影响。

（3）控制方向可选定：对正向操作（冷却），当 PV 值增加时控制变量是增加的；对反向操作（加热），当 PV 减小时控制变量增加。如图 4-69 所示。

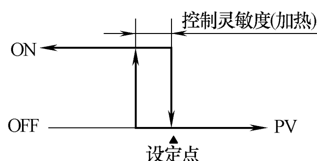


图 4-68　灵敏度设定

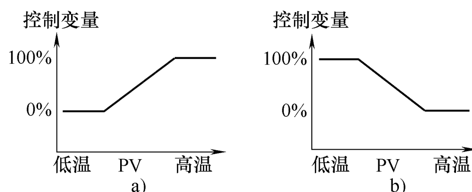


图 4-69　正、反向控制

a）正向控制　b）反向控制

当用反向 ON/OFF 控制时，当 PV 值低于 SV 值时控制输出为 ON。当 PV 等于或大于 SV 时控制输出为 OFF。当用正向 ON/OFF 控制时，情况相反。

用单元 DIP 开关的第 4、5 针可以将温度控制单元的控制方向设定为反向操作或正向操作，出厂时设定为反向操作（加热）。

（4）数据格式可选定：可用单元前面板上的 DIP 开关（第 3 针），选择温度控制单元的数据，是用 4 位 BCD 码，还是用二进制（即 4 位十六进制）码处理。

（5）有输入补偿功能：对传感器测得的温度值，可增加一个补偿值，来调整 PV 值。

（6）加热器烧断检测（仅用于单相操作）：当使用 2 回路温度控制单元时，每个回路可以接一个电流互感器（CT）以检测加热器烧断。

（7）每个回路有两个内部报警，可将报警输出到 CPU 单元的存储区中分配的区域，而且可以使用下列 9 种报警模式的任一种：高和低限报警、高限报警、低限报警、带备用顺序的高低限报警、带备用顺序的高限报警、带备用顺序的低限报警、绝对值高限报警、绝对值低限报警。

报警输出可设成有迟滞的。报警迟滞与报警输出 ON/OFF 间的关系，如图 4-70 所示。

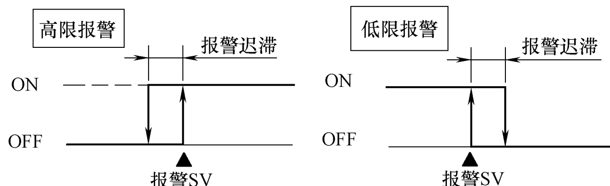


图 4-70　报警迟滞与报警输出 ON/OFF 间的关系

报警还有个“待机顺序”问题。此即：在单元初始化期间，其 PV 进入报警范围时，待机顺序禁止报警输出。等 PV 值离开报警范围，下一次再进入报警范围后，报警输出将起作用。但待机顺序可重新起动，如单元重新起动时或当输出，OFF 又转为 OFF 时，即又进入待机顺序状态。

（8）与 CPU 单元可交换多种数据：其交换数据的情况如图 4-71 所示。

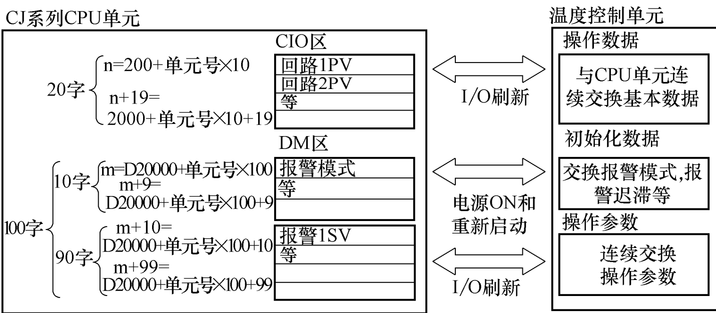


图 4-71　交换数据简图

数据交换通过 CPU 单元的 CIO 和 DM 区中的字执行。操作数据存储在 CIO 区，初始化数据和操作参数存储在 DM 区。

操作数据为操作温度控制单元用的基本数据，在 CPU 单元 I/O 刷新时作交换。操作数据包括过程值、设定点、停止位、AT 起动位、AT 停止位和其它数据。所以，可从 CPU 单元用操作数据，向温度控制单元发命令，以切换温度控制单元 PID 控制的运行和停止。但当 PLC 处编程模式时，可用单元前面板上的 DIP 开关（第 1 针），选择温度控制单元是继续操作还是停止操作。

初始化数据用于初始化温度控制单元的数据，是在 PLC 转为 ON 或温度控制单元重新起动时，作为初始化数据与 CPU 单元交换的，初始化数据包括报警模式，报警迟滞和其它数据。

操作参数为控制温度控制单元操作的参数，是在 CPU 单元 I/O 刷新时，作为操作参数与 CPU 单元交换的。操作参数包括报警 SV、控制周期、比例带、积分时间和其它参数。

（9）温度控制单元有自己的数据存储器：

温度控制单元有两种存储器：RAM 和 EEPROM。表 4-13 说明了这些数据存储与传送的情况。

表 4-13　数据存储与传送

<table>
<tr><th colspan="2">CPU 单元中存储器分配</th><th>主要设定</th><th>从 CPU 单元区传送到温度控制单元 RAM</th><th>从 RAM 传送到 EEPROM</th><th>从温度控制单元中的 EEPROM 传送到 CPU 单元区</th></tr>
<tr><td>CIO 区</td><td>操作数据</td><td>设定点
加热器烧断电流</td><td>I/O 刷新周期</td><td colspan="2">不传送</td></tr>
<tr><td rowspan="3">DM 区</td><td>初始化数据</td><td>报警模式
报警迟滞</td><td>电源 ON 或单元重新起动</td><td colspan="2">不传送</td></tr>
<tr><td rowspan="2">操作参数</td><td>报警 SV
输入补偿
控制周期
灵敏度</td><td>I/O 刷新周期</td><td rowspan="2">当特殊 I/O 单元区中保存位转为 ON 时</td><td rowspan="2">当 DIP 开关上的第 8 针转为 ON，并且电源转为 ON 或单元重新起动时</td></tr>
<tr><td>比例带
积分时间
微分时间</td><td>I/O 刷新周期和 PID 常数的变化标志为 OFF 时一样长（见注）</td></tr>
</table>

注：1. 自动整定的 PID 参数时，其结果在自动整定后自动写入 RAM。
2. EEPROM 写入的寿命为 100000 次。

温度控制单元的数据是，在与 CPU 单元交换数据时，写到温度控制单元 RAM 中的。如保存位置为 ON，可以将这些数据的一部分从 RAM 写到 EEPROM。如果 DIP 开关的第 8 针为 ON，当电源转为 ON 或重新起动温度控制单元时，储存在 EEPROM 中的数据将自动传送到 CPU 单元的 DM 区，以便能用储存在 EEPROM 中的数据操作。图 4-72 说明过了这些数据传送的情况。

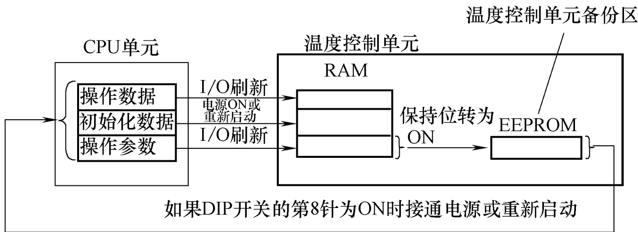


图 4-72　数据传送概况

所以，为了简化温度控制单元的操作，可以将 DIP 开关的第 8 针转为 ON，可方便地用储存在 EEPROM 中的数据操作。

（10）可使用 OMRON 公司改进的 PID 算法，实现双自由度控制，以提高控制的性能。而且，其 PID 参数是可自整定的。

4. 使用

温度控制单元的安装和设定步骤如下。

（1）设定单元号。用面板上的单元号开关，设定单元号。单元号为两位数。设定后，在 CIO 区占用 20 字，在 DM 区占用 100 字。单元号与使用的 CIO 起始地址的对应关系为

$$n = 200 + (10 \times 单元号)$$

共要用 20 个字，从 $n \sim n+19$。

（2）设定输入类型。可用面板上的输入类型开关，选定在温度控制单元的输入类型。其间关系见表 4-14（用于热偶）及表 4-15（用于热阻）。

表 4-14　输入类型开设定（用于热偶）

输入类型设定	类　　型	温　度　范　围	
		摄　　氏	华　　氏
0	K	-200 ~ 1300℃	-300 ~ 2300℉
1	K	0.0 ~ 500.0℃	0.0 ~ 900.0℉
2	J	-100 ~ 850℃	-100 ~ 1500℉
3	J	0.0 ~ 400.0℃	0.0 ~ 750.0℉
4	T	-200.0 ~ 400.0℃	-300.0 ~ 700.0℉
5	L	-100 ~ 850℃	-100 ~ 1500℉
6	L	0.0 ~ 400.0℃	0.0 ~ 750.0℉
7	R	0 ~ 1700℃	0 ~ 3000℉
8	S	0 ~ 1700℃	0 ~ 3000℉
9	B	100 ~ 1800℃	300 ~ 3200℉

表 4-15　输入类型开设定（用于热阻）

输入类型设定	类　型	温 度 范 围	
		摄　氏	华　氏
0	Pt100	-200.0~650.0℃	-300.0~1200.0℉
1	JPt100	-200.0~650.0℃	-300.0~1200.0℉
2~9	不设定 2~9		

例：对热偶型的单元，如将单元前面板上的开关设定到 1，则输入为 K 型热电偶（0.0~500.0 ℃）。

（3）设定控制功能。用面板上的 DIP 开关（图 4-73，在右边为 ON），按要求作控制功能的设定。DIP 开关各针的作用见表 4-16。

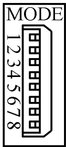


图 4-73　DIP 开关

例：如针 2 置 OFF，则选择的温度单位为摄氏度；而针 3 置为 OFF，则选择的数据格式为 BCD 码。等等。

表 4-16　DIP 开关设定

针	功　能	ON	OFF	出厂设定
1	当 CPU 单元处于编程模式时的操作	继续	停止	OFF · · ·
2	温度单位(℃/℉)	℉	℃	
3	数据格式	16 位二进制	4 位 BCD	
4	控制操作(回路 1 和 3)	正向 (冷却)	反向 (加热)	
5	控制操作(回路 2 和 4)	正向 (冷却)	反向 (加热)	
6	控制方式	ON/OFF 控制	PID 控制	
7	EEPROM 中的初始化设定值	初始化	不初始化	
8	传递 EEPROM 中的设定值	传送	不传送	ON

（4）按要求接线。

（5）接通 PLC 电源。

（6）创建 I/O 表。

（7）在分配的 DM 区内分配给单元的字中进行初始设定。

（8）再做一次（断开再合上）PLC 供电循环，使设定生效。

（9）编程。

按要求编写有关程序。如单元号设为 1，要求编一个把过程值（PV）分别存储到 D100~D103 中的程序。

由于本例单元号为 1，回路 1、2、3、4 的 PV 值的地址分别为 2013、2014、2023 及 2024。回路传感器出错标志位分别为 201814、201914、202814 及 202914。只要相应的传感器不出错，就要把有关数据予以存储。这程序是很简单的，如图4-74所示。



图 4-74　存储程序

（10）PID 参数整定。

PID 参数可手动设定，也可自动整定。

1）手动设定。较简单，按要求，将比例带（P），积分时间（I）和微分时间（D）等值，设定到分配给单元的 DM 字中，即可。

2）自动整定。自动整定使用的是有限周期法，通过强制改变操作变量来测定控制系统的特性，以确定最佳的 PID 常数。

自整定起动由 PLC 控制。起动后的计算由单元自行实现。求得自整定的参数后传送、保存还由 PLC 控制。

具体步骤是：

（a）将 AT 起动位，如回路 1 使 n+2 的 02 位转为 ON，起动自动整定。

（b）单元进行自动整定。

（c）自整定完成时，将求得的 PID 参数储存到分配给该单元的 DM 字内的操作参数输入区（求得的 PID 参数从温度控制单元传送到 CPU 单元）。

所以，要进行自整定，要有相应的 PLC 程序。图 4-75 即为一个程序实例。它用的单元为 CJ1W-TC001 温度控制单元。单元号设为 0，用于回路 1 的 PID 参数自整定。

注：1. 回路 1 AT 标志为 CIO（n+8）的位 03，回路 1 PID 常数计算标志为 CIO（n+8）的位 10，而回路 1 保存完成标志为 CIO（n+8）的位 15。

2. 回路 1AT 起动位为 CIO（n+2）的位 02 而回路 1 改变 PID 常数位为 CIO（n+2）的位 13。

3. 当改变 PID 常数位转为 ON 时，PID 常数计算标志将为 OFF。

4. 如果 DIP 开关的第 8 针设定为 ON，则在初始化时单元的 EEPROM 中的设定传送到 CPU 单元，必须将回路的保存位转为 ON 以将新的设定保存到温度控制单元的 EEPROM。

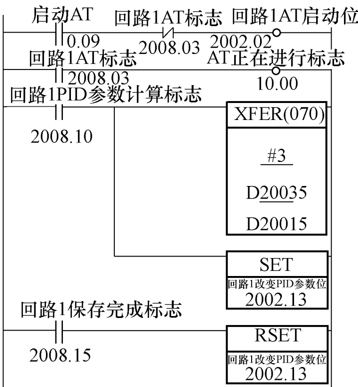


图 4-75　自整定程序实例

4.7.3　回路控制单元

回路控制单元（LC）或回路控制内插板（LCB）主要用于回路 PID 控制。前者结构与其它特殊单元相同，最多的可控制 32 个回路。后者嵌入 CPU 单元中，故称内插板，具有更强的功能，最多可控制 500 个回路。这两者，除了用于回路控制，还可进行顺序逻辑控制及步进控制。也可不做控制，而单独用作 PLC 的报警与监控终端。

OMRON 公司开发此类模块，用的是基于 PLC 的过程控制技术，以在 PLC 平台上，实现过程控制。由于它保留了传统 PLC 特点，所以，用它配置的过程控制系统，与 DCS 相比，具有高得多的性能价格比。

图 4-76 所示为该单元的外观图。表 4-17 示出回路控制单元（板）的型号与规格。

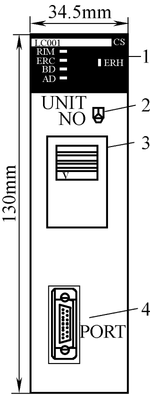


图 4-76　回路控制单元

1—指示灯　2—单元号开关

3—电池盒　4—RS-232 通信口

1. 特点

（1）该单元无硬件输入、输出通道或输入、输出接点，仅有各种功能很强的软件模块。所以，实质上它只是协助 PLC CPU 处理过程控制的协处理器。但是，通过自身的终端等功能块（也是软件模块），它可访问与 PLC 输入、输出通道对应的内部器件或 PLC 其它内存区。通过自身的接口

表 4-17　回路控制单元（板）型号

名　　称	规　　格	型　　号
回路控制板	功能块数量:50 块 max	CS1W-LCB01
	功能块数量:500 块 max	CS1W-LCB05
	功能块数量:500 块 max 支持双机(CS1D)操作	CS1D-LCB05D
回路控制单元	控制回路数:32 回路 max 操作数量:249max	CS1W-LC001

功能块（也是软件模块），还可与在网络节点上的其它 PLC 的交换数据。所以，用它进行回路控制或实现其它功能，需要模入、模出等单元或网络模块配合；也可能得到模入、模出等单元或网络模块配合。

（2）该单元的软件功能模块有：

1）系统公用（System Common）块，编号 000，用于所有功能块的公共设定及系统输出。

2）各种控制（Control）块，如：两位置 ON/OFF 控制，编号 001，用于两位置 ON/OFF 控制，块地址 000。三位置 ON/OFF 控制，编号 002，用于三位置 ON/OFF 控制，块地址从 001 开始。基本 PID 控制，编号 011，用于基本 PID 控制，块地址从 001 开始。改进 PID 控制，编号 012，具有 OMRON 前馈控制的 PID 控制，块地址从 001 开始。混合 PID 控制，编号 013，用于累计量控制，块地址从 001 开始。流量控制，编号 014，用于是否到达设定流量值，控制阀门启闭，块地址从 001 开始。模糊逻辑，编号 016，用于模糊控制，块地址从 001 开始。显示及设定，编号 031，用于现值显示及给定值设定，块地址从 001 开始。显示及操作，编号 032，用于现值显视及输出值设定，块地址从 001 开始。比率设定，编号 033，用于现值显示及比率与偏差设定，块地址从 001 开始。现值指示，编号 034，用于带报警的现值指示，块地址从 001 开始，等等。

3）接口（External Controller）块。

4）处理（Operation）块。

5）顺序控制（Sequential　Control）块。

6）终端（Field Terminal）块，等等。

（3）实现该模块的功能，就是靠运行它自身的程序，调用这些软件模块。此程序是独立于 PLC 程序而自行运行的。其循环运行的周期（控制周期）可用工具软件设定，与 PLC 的扫描周期无关。周期与控制的路数有关。周期小，则控制的回路数少。而且，只要上电，模块的程序就可运行，或由工具或监控软件控制其运行。而与 PLC 处于什么工作模式无关。

（4）不能用通用的 PLC 编程软件编程。要用 OMRON 提供的回路控制模块的编程工具软件（CX-Process Tool）编程。该软件是可视化的，编程很简单。就像建立流程图一样，先选定好功能块，然后用鼠标将这些块连连接线，相互连接起来，再作些参数选定就可以了。可以进行从基本 PID 控制到串级和前馈控制等各种各样的控制。增加控制回路的数量，仅是增加模块调用。但编程后，要通过 HOST LINK 网（串口）或 CONTROL LINK 网的通信口下载给模块，才能生效。

（5）该模块无人机界面，要用 OMRON 提供的回路控制模块的监控软件（CX-Process Monitor）进行数据设定与监视。CX-Process Monitor 是可视化的软件，可实现图形监视，可建趋势图，报警记录等。也可用其它可视化的监控软件进行数据设定与监视，以建立友好的人机界面。

2. 使用

以下举一个控制用回路控制板，实现液槽的串级 PID 控制实例，说明它使用的大体步骤。

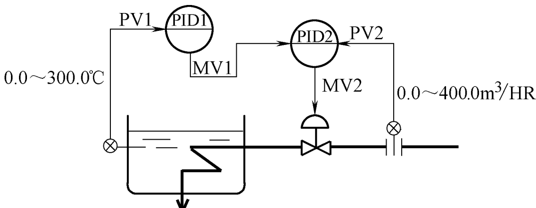


图 4-77　两个 PID 控制回路

（1）设计。

1）画原理图。

图 4-77 示的就是系统的原理图。它有两个 PID 回路，PID1、PID2。PID1 为外环，控制液槽的温度，它的输出值为 MV1，作为 PID2 的设定值。PID2 控制加热蒸汽的流量。

2）PLC 系统的配置。

主要考虑图 4-77 的需要，系统配置如图 4-78 所示。

3）选择功能块。

按系统工作要求，选用的模块有：一个模入块，一个模出块，一个开方运算块及两个基本 PID 块。其关系如图 4-79 所示。

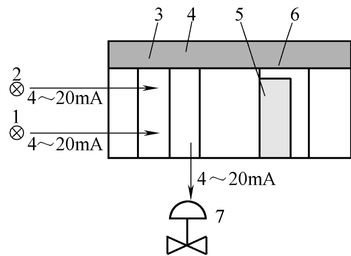


图 4-78　PLC 系统的配置

1—压力变送器　2—温度变送器　3—模入单元　4—模出单元　5—回路控制板　6—CPU 单元　7—调节阀

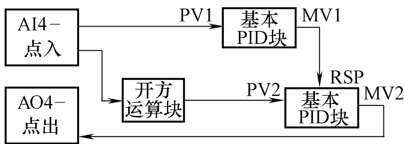


图 4-79　关系图

4）决定功能块系统配置。

5）用 SCADA 软件决定数据监控。表 4-18 所示为有关监控的设定。

表 4-18　监控设定

块地址	功能块名	标签号	标签注解	报警低限	报警高限	单位
001	基本 PID	PID1		0	300	℃
002	基本 PID	PID2		0	400	m^3/HR

（2）用 CX-Process Tool 设定功能块数据。

1）运行 CX-Process Tool 软件。

2）设定系统公用块（功能模块 000）数据。在此设定有关模块工作的初始化数据。如本例，软件设定画面如图 4-80 所示。所设的数据见表 4-19。

3）用 CX-Process Tool 软件，选择要用的功能块及其地址分配，如图 4-81 所示。

4）在模块之间进行软件连线，如图 4-82 所示。

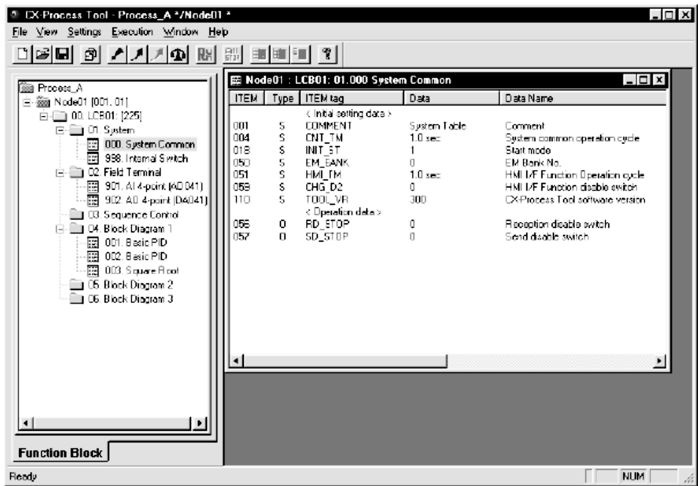


图 4-80　软件设定画面

表 4-19　设定数据

项　　目	数　据　说　明	数据范围	设定实例
004	控制周期(s) 1:0.1,2:0.2,3:0.5,4:1,5:2	1 ~5	3
018	上电时起始模式 1:热起动 2:冷起动	1 或 2	1

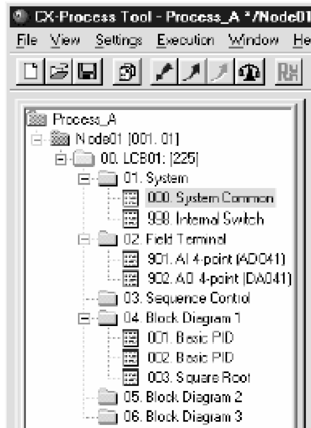


图 4-81　功能块选择及地址分配

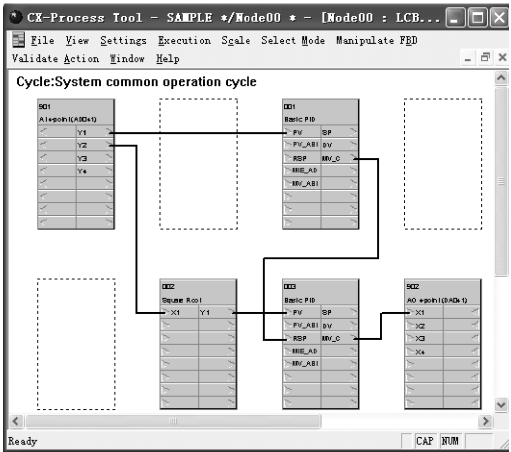


图 4-82　软件连线

5）对每个模块的项目进行设定，如图 4-83 所示。

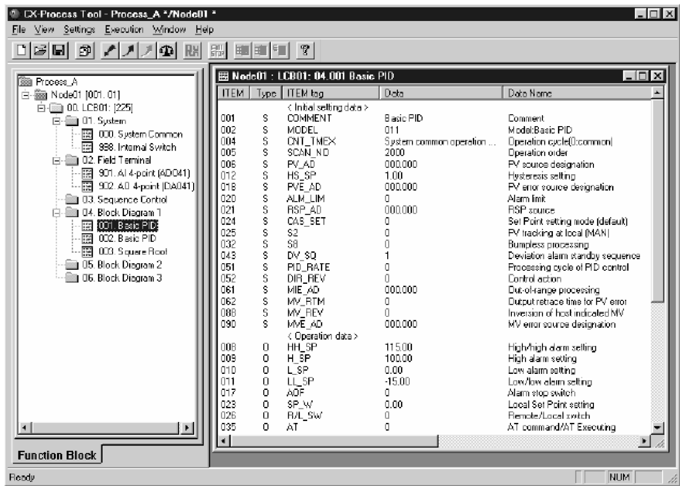


图 4-83　模块项目设定

（3）回路控制板及有关设定。

1）安装回路控制板及做好模入、模出单元接线，如图 4-84 所示。但回路控制板无须接线。

2）连接编程器。

3）接通电源。

4）用编程器建 I/O 表。

5）如需要，作通信口设定。

6）设定模入、模出单元地址。

（4）下载功能块数据给回路控制板。

1）关断 PLC 电源。

2）对 CPU 单元面板上的 DIP 开关进行设定（SW4：ON ，当使用外设口时；OFF ，当使用 RS-232C 口）。

3）PLC CPU 与计算机连接，如图 4-85 所示（也可用外设口，推荐电缆为 CS1W-CN226/626）。并运行 CX-Process Tool 软件。

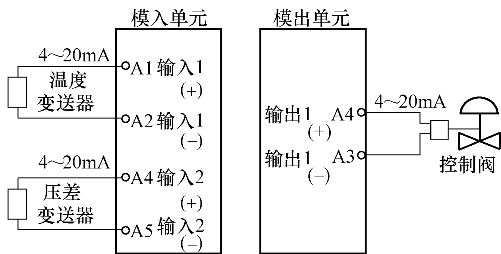


图 4-84　模入、模出单元接线

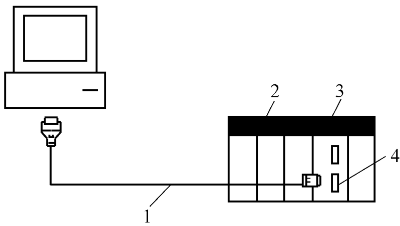


图 4-85　与计算机连接

1—连接电缆　2—回路控制板　3—CPU 单元　4—RS-232 通信口

4）接通电源。

5）在 CX-Process Tool 上，设定网络地址（000）及节点地址（01）。

6）在 CX-Process Tool 上，建立 Host Link 连接。

7）下载在 CX-Process Tool 上准备好的功能块数据给回路控制板。

8）在 CX-Process Tool 上，执行 run/stop 命令或使 PLC 断电再加电。

9）检查回路控制板的面板指示灯。应该是：RUN 亮；而 ERC 不亮。

（5）跟踪操作。

1）在 CX-Process Tool 上，执行 run/stop 命令或使 PLC 断电再加电。

2）在 CX-Process Tool 上，检查系统运行。进行负荷检查及其它诊断。

3）设定并起动 CX-Process Tool 或监控软件。

4）在 CX-Process Tool 上或 SCADA 软件上，设定给定值及其它设定。

（6）实际操作。

1）用 SCADA 软件操作回路控制板，如改变给定值或 PID 参数。

2）用 SCADA 软件监控调节量现值及报警。

4.8　PID 控制高级应用

关键词：PID 串级控制、多回路交叉 PID 串级控制、交叉限幅 PID 串级控制、前馈与 PID 混合控制

PID 指令或函数块，不仅可简单地用于单个回路的 PID 控制，也可用于串级控制等高级 PID 控制。

4.8.1　串级 PID 控制

串级 PID 控制有主、辅两个控制回路，现以某煤气燃烧炉的温度控制为例对其作说明。图 4-86 所示为该系统的简图。

从图知，它用了两个 PID 函数块，主 PID 函数块（控制温度）及辅 PID 函数块（控制流量）。

主 PID 函数块 T 反馈输入信号为“实际温度”，它是把检测到的加热炉温度，经输入转换得来的。主 PID 函数块的给定值为“给定温度”，按加热炉的温度要求选定。主 PID 函数块的控制输出为燃气“给定流量”。用它作辅助控制回路的给定值。

辅 PID 函数块 G 反馈输入信号为“实际流量”，它是把检测到的燃气管道中的流量，经输入转换得来的。主 PID 函数块的控制输出“给定流量”，即为辅 PID 的设定值。辅 PID 函数块的控制输出，经输出转换，加载给调节阀，以确保有合适流量的燃气流入加热炉。

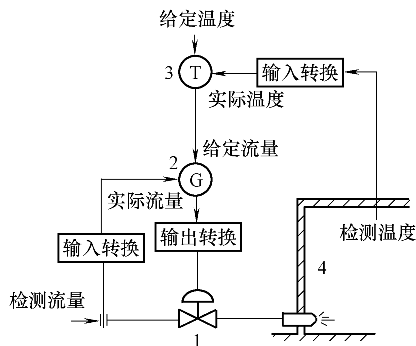


图 4-86　串级 PID 控制简图
1—调节阀　2—辅 PID 函数块
3—主 PID 函数块　4—加热炉

从 PID 函数控制原理可知，当加热炉“实际温度”低于“给定温度”时，主 PID 函数块的控制输

出，“给定流量”将增加。即辅 PID 函数块的给定流量增加。这时，如实际流量没有变化，则流量偏差将增大。经辅 PID 函数块调节，将增大对调节阀的控制输出，使阀门开大，进而，增加燃气流量。

显然，燃气流量增加，将使加热炉温度上升，即实现对温度的调节。

这里，由于可使所控制的炉温免受或少受燃气压力波动的干扰，从而提高系统的控制品质。

4.8.2 串级比例双辅助回路 PID 控制

图 4-87 为串级双辅助回路 PID 串级比例控制示意图。主控回路控制温度，两个辅助控回路，一个控制燃气流量，另一个控制空气流量。它也是某煤气燃烧炉的温度控制的实例。

从图知，它用了 3 个 PID 函数块，一个主 PID 函数块 T（控制温度）及两个辅 PID 函数块 G_1、G_2（一个控制燃气流量，另一个控制空气流量）。

主 PID 函数块反馈输入信号为“实际温度”，它从检测到的加热炉温度，经输入转换得来。主 PID 函数块的给定值为“给定温度”，按加热炉的温度要求选定。主 PID 函数块控制输出为“给定燃气流量”。用它作两辅控制回路的给定值。

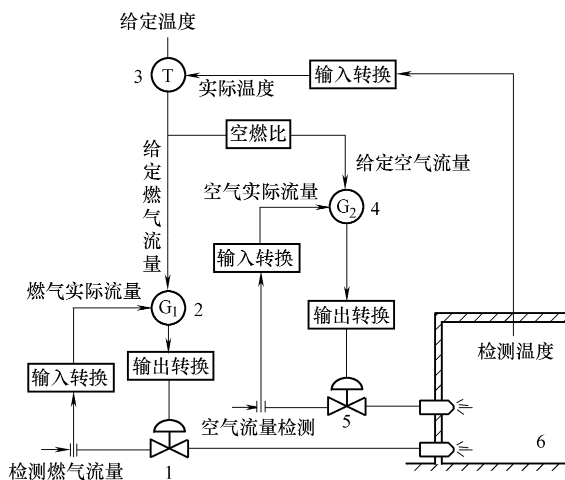


图 4-87 串级比例双辅助回路 PID 控制示意图
1—燃气调节阀 2—燃气 PID 函数块 3—温度 PID 函数块 4—空气 PID 函数块 5—空气调节阀 6—加热炉

燃气控制辅 PID 函数块反馈输入信号为“实际燃气流量”，它从检测燃气管道中的流量，经输入转换得来。主 PID 函数块输出的“给定燃气流量”，即为它的给定值。燃气控制辅 PID 函数块的控制输出，经输出转换，加载给燃气调节阀，以确保有合适燃气流量的燃气流入加热炉。

空气控制辅 PID 函数块反馈输入信号为“实际燃气流量”，它从检测空气管道中的流量，经输入转换得来。主 PID 函数块控制输出的“给定燃气流量”，与“空燃比”（单位重量的燃气完全燃烧所需要的空气重量）相乘，即为它的给定值。空气控制辅 PID 函数块的控制输出，经输出转换，加载给空气调节阀，以确保有合适空气流量的空气流入加热炉。

这里空气控制回路的设定值，总是与燃气控制回路的设定值，按“空燃比”规定的比例变化，目的是使在加热炉中燃气与空气保持合适的比例，以保证有较高的燃烧效率。这是生产工艺对控制系统的要求，也是控制系统必须保证的。

从 PID 函数控制原理可知，当加热炉“实际温度”低于“给定温度”时，主 PID 函数块的控制输出的“给定燃气流量”将增加。即燃气控制 PID 函数块的给定燃气流量增加。如这时的实际燃气流量未变化，则燃气流量偏差增大。经燃气控制辅 PID 函数块调节，将增大对燃气调节阀的控制输出，使燃气调节阀阀门开大。进而，增加燃气流量。同时，空气控制

辅 PID 函数块的给定流量也将按比例（为“空燃比”的倍数）增加。如这时的实际空气流量没有变化，则空气流量偏差增大。经空气控制辅助 PID 函数块调节，将增大对空气调节阀的控制输出，使空气调节阀阀门开大。进而，增加空气流量。

显然，燃气流量、空气流量按比例地都增加，既可保证加热炉良好燃烧，又将使加热炉温度上升，也就较理想地实现了对温度的调节。

4.8.3　串级比例并交叉限幅双辅回路 PID 控制

图 4-88 为串级比例并交叉限幅双辅助回路 PID 控制示意图。主控回路控制温度（见图 4-88，这里略），两个辅助控制回路，一个控制燃气流量，另一个控制空气流量。它也是用于某煤气燃烧炉的温度控制。

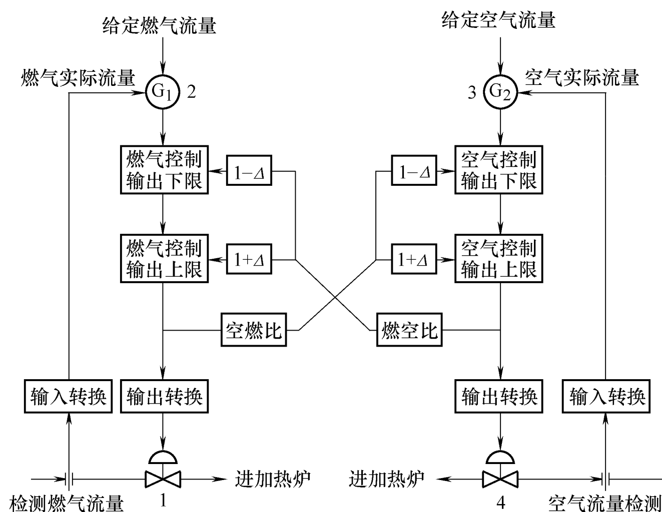


图 4-88　串级比例并交叉限幅双辅助回路 PID 控制示意图
1—燃气调节阀　2—燃气 PID 函数块　3—空气 PID 函数块　4—空气调节阀

从图知，这里函数块 G_1、G_2 的使用与图 4-87 不同的只是，都起用了上下限限幅功能。限幅的含义是：当控制输出大于设定的限幅上限时，就按上限值输出；当控制输出小于设定的限幅下限时，就按下限值输出。这里控制输出的上下限，由对方控制输出乘“燃空比”（燃空比为空燃比的倒数），再乘“$1+\Delta$”（用于上限），或再乘“$1-\Delta$”（用于下限）得到。

为什么要作这个限幅呢？因为有了这个限幅，尽管这两个辅助回路动态特性不完全相同，但仍可保证，在温度调整的过渡过程中，燃气、空气流量的比例与理想的比例相差不大，以做到不仅在稳态时，燃烧效率高，而且，在过渡过程，燃烧效率也高。

图 4-87 与图 4-88 组合，即为实际的用于某煤气燃烧炉的温度控制。运行证明，控制效果是很好的，既节约了能量，又保证了工艺质量，是未来燃气加热炉控制的方向。

4.8.4　前馈与 PID 混合控制

前馈控制可与 PID 控制结合使用，以提高控制效果。这里，前馈用于抗干扰（针对一些突然变化的参数），而 PID 用于常态控制。这在煤炭加热炉的炉膛压力控制中，很常用。

图 4-89 所示为这个应用。这里，为了保持炉膛出口压力处于给定的微负压，用 PID 算法控制引风机转速。当出口压力增大，引风机转速提高，加大引风量，将使出口压力降低。反之，引风机转速降低，减少引风量，将使出口压力提高。

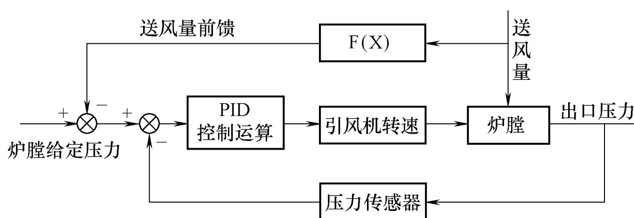


图 4-89 炉膛压力控制简图

但是，在燃烧过程中，为了保持温度，送风量将随着煤炭输送量的变化经常有所变化。而这个送风量变化是出口压力的主要干扰量，为此，可对此量作前馈控制处理。

从图知，它的 PID 控制运算的给定值为"炉膛给定压力"与"送风量前馈"之差。而"送风量前馈"的 F（X）多为与"送风量"微分成正比的值。实质上这也是 PID，只是比例、积分控制作用令其为 0，而只保留微分作用。

显然，有了这个前馈，当送风量突然增加，将使加载给 PID 控制运算的输入减少，使炉膛压力骤减，而这时送风量增多，则将使炉膛压力骤增。处理得好，可使两者得以相抵，可减少出口压力的波动。

以上讲的是算法，而在程序上只是根据此算法，使用好 PID 指令及做好指令使用的前后处理就可以了。具体程序略。

4.9 模糊控制

关键词：模糊性、模糊集、隶属度、模糊控制、模糊化、多维模糊控制、多变量模糊控制、解模糊、模糊推理

模糊控制，英文称 Fuzzy Control，与以上讨论的模拟量控制，其输出与输入关系总是精确的不同，其输出与输入关系是模糊的。它基于人们的经验总结出来的一系列规则，并运用这些规则归纳出的算法去实现控制。

在实际工程中，许多系统和过程都十分复杂，难以建立确切的数学模型和设计出通常意义下的控制程序。只能由熟练操作者凭借经验以手动方式控制。其控制规则常常以模糊的形式体现在控制人员的经验中，很难用传统的数学语言来描述。这时，使用模糊控制就有可能获得满意的控制效果。

4.9.1 模糊控制原理

1965 年美国的控制论专家札德（L. A. Zadeh）教授提出模糊性概念，创立了模糊集合论。又在 1968 ~ 1973 年期间先后提出语言变量、模糊条件语句和模糊算法等概念和方法，使得某些以往只能用自然语言的条件语句形式描述的手动控制规则可采用模糊条件语句形式来描述，从而使这些规则成为在计算机上可以实现的算法。

1974 年 E. H. 曼达尼和 S. 阿西里安成功地把这种想法应用于小型汽轮机的控制，开拓了模糊控制的方向。此后，模糊控制方法迅速得到推广，被应用于热交换器、水泥窑、交通管理等许多领域。

PLC 模糊控制是一系列模糊控制技术的一种。它的核心是利用模糊集合理论，把人的控制策略的自然语言，转化为 PLC 的知识库（使用模糊控制单元时，建立的数据库存于该单元的内存中）及程序，以便在 PLC 运行程序时，能模拟人的思维方式，对一些无法构造数学模型的被控对象进行有效的控制。

PLC 模糊控制目前用的还不算太多。专用的控制单元，见到的似乎仅 OMRON 等少数公司才有。但是，即使无这种特殊单元，用模糊控制的算法，去设计 PLC 控制程序则是不受限制的。图 4-90 示出模糊控制的基本原理。

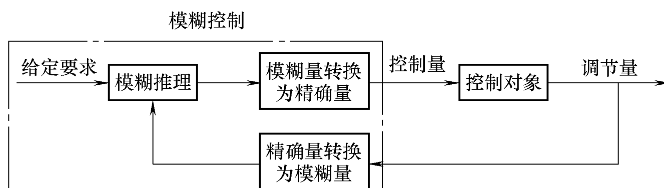


图 4-90　模糊控制原理图

从图知，模糊控制也是闭环控制，它也要不停地检测控制对象的输出（调节量）。但它要把输入这个精确量，转换为模糊量（称模糊化），进而利用输入输出间的模糊关系进行模糊推理。模糊关系就是基于人们，特别是专家的经验，所形成的一系列规则（大前提）。模糊推理把检测到的模糊输入（小前提）与模糊关系结合进行判断，以给出控制对象应得到的控制。只是这个结论也是模糊的。要把这个结论，用作控制量，还需把它转换为精确量（称解模糊）。

可知，这里模糊控制就是根据系统输出的误差和误差变化情况来决定控制量。在手工操作情况下，这项工作原来是由控制人员通过手动控制完成的。把他们的经验表述为一套自然语言的条件语句，再应用模糊集合论将其转化为一组模糊条件语句，就可用来组成模糊控制规则。例如，对于由下述语句表述的经验规则：“如果误差很大，且误差继续朝不利方向很快变化，应加大控制量；如果误差大小为中等程度，且朝着有利于减小误差的方向变化，应使用很小的控制量来使误差继续缄小，……”。

模糊控制的特点是不需要考虑控制对象的数学模型和复杂情况，而仅依据由操作人员经验所制定的控制规则就可构成。凡是可用手动方式控制的系统，一般都可通过模糊控制方法设计出由 PLC 可执行的模糊控制程序。模糊控制所依据的控制规律不是精确的。其模糊关系的运算法则、输入精确量到模糊量的转换，以及输出量模糊量到实际的控制量的转换等，都带有相当大的任意性。对于模糊控制的性能和稳定性，常常难以从理论上作出确定的估计，只能根据实际效果评价其优劣。

以下举一个洗澡水温度控制实例，看看模糊控制是怎样进行的。

洗热水澡时，洗澡水一般应是“暖和”的。暖和与不暖和，虽然可用洗澡水的温度衡量，但不好硬性地说，多少度到多少度之间的洗澡水是暖和的，而其它温度的洗澡水就不是暖和的。因为，洗澡水这个暖和的温度特征不是很清晰的，而是模糊的。

所以，在不同温度的洗澡水中，哪些属于暖和，哪些属于不暖和，不好用 17 世纪康德创立的集合论那样，去精确地划分。而要用扎德提出模糊集合的概念处理。

在这里，扎德引入“度”的概念。这个“度”不是温度的度，而是模糊集合中隶属度

的度（Grade）。对某人、某季节，可这样假设，如 40℃时，用作洗澡水正好是“暖和”的，则设其隶属度为 1. 0（百分之百）。而 35℃，或 45℃呢？就不那么正好了，设其隶属度设为 0. 5（百分之 50）。低于 20℃就太凉了，设其隶属度设为 0。而高于 60℃就太热了，也设其隶属度也设为 0。有了这几个特殊温度点隶属度的假设，其它温度时，洗澡水属于暖和的隶属度则可从图 4-91 得知了。

有了这个图，即可把不同温度的洗澡水，本来是精确量的温度量，变换为使人们能予以感觉的模糊量，即模糊子集“暖和”，并用隶属度反映其模糊程度。

显然，这个从温度的度到这里的“度”的转换，完全是凭经验得出的。

除了“暖和”，还有“太凉”、“凉”、“热”、“太热”等也都是模糊的概念，也可有对应的模糊子集。其隶属度分别与温度的关系如图 4-92 所示。

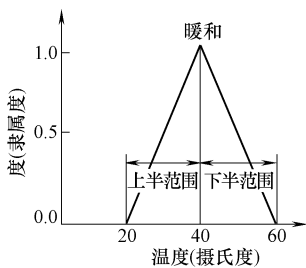


图 4-91 “暖和”的隶属度

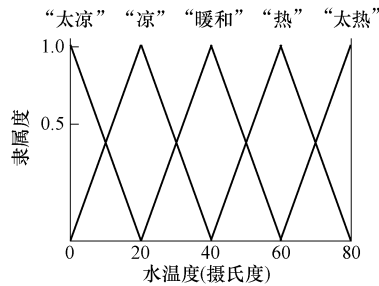


图 4-92 “太凉”、“凉”、“热”、“太热”的隶属度

这里各模糊子集的隶属度与温度的关系都是线性的，分别为“Δ”，或半“Δ”型。也可为别的形，到底怎样，可依经验确定。

有了图 4-91 及图 4-92，可很容易地把有精确温度值的水的状态，转换为一些，如“暖和”、“凉”、“热”……模糊子集，精确量的模糊化就有依据了。

要使洗澡水达到“暖和”的要求，可依水的状态，凭经验是，拧动水龙头。大体是（为了简化，这里未考虑水的变化的情况）：

洗澡水“太热”了，水龙头较多地拧向凉水方（ZLL）；

洗澡水“热”了，水龙头拧向凉水方（ZL）；

洗澡水正好“暖和”，水龙头不动（ZO）；

洗澡水“凉”了，水龙头拧向热水方（ZR）；

洗澡水“太凉”了，水龙头较多地拧向热水方（ZRR）。

这里“水龙头较多地拧向热水方”等结论也是模糊的。它与输出水龙头转角这个精确量间的关系，如图 4-93 所示。

这里各模糊集合的隶属度与输出水龙头转角的关系是垂直形。也可为别的形，到底怎样，也是依经验确定。

有了这些，可以看看，怎样从检测到的实际温度，去控制水龙头的相对转角的。

如检测的温度这正好是 40℃，那只有“暖和”的隶属度为 1，其它的均为 0。依上述规则，其结论，应为“水龙头不动”（隶属度为 1）。而“水龙头不动”（隶属度为 1）对应的相对转角正好 0。

如检测的温度这为是 35℃，那“暖和”的隶属度为 0. 75，“凉”的隶属度为 0. 25，其

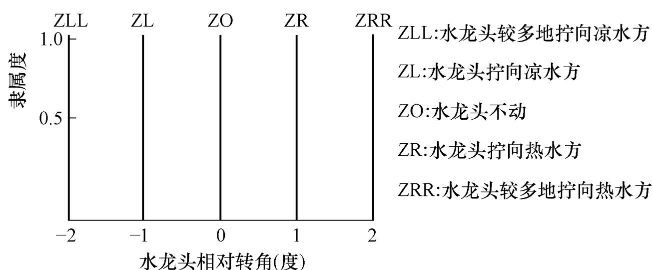


图 4-93　水龙头拧向隶属度

它的均为 0。依上述规则，其结论，应为“水龙头不动”，隶属度为 0.75；“水龙头拧向热水方”，隶属度为 0.25。这时转角多少呢？

有两个处理方法：

（1）选隶属度大者输出。如本例“水龙头不动”，隶属度大。选它，其对应的则是水龙头相对转角 0。

（2）依隶属度加权平均确定输出。即：

水龙头相对转角 = 0.75 × 0 + 0.25 × 1 = 0.25（度）

从以上介绍可知，不管检测到的温度是多少，总可计算出与其对应的水龙头相对转角。因此，用这种模糊控制原理是可实现控制的。至于控制的效果如何，正如 PID 控制要选好 PID 参数那样，也要确定好输入量到模糊集的转换、控制规则及模糊量到输出的转换。

提示：模糊控制没有用什么精确的公式，也没有什么微分方程。是基于从经验总结出的语言型规则，进行控制。其控制机理及策略很易理解，设计简单，应用简便。

4.9.2　模糊控制算法

从模糊控制原理知，模糊控制算法的要点是：从输入到模糊量的转换，也称模糊化；建立控制规则，进行模糊推理；从模糊量到输出的转换，也称解模糊（Defuzzification）。

1. 模糊化算法

输入量模糊化的任务是，把检测到的被控量值转换为相应隶属度的模糊子集。为此，首先要对输入量（被控重）划分模糊集。如上例输入量有五个模糊子集，即：“太凉”、“凉”、“暖和”、“热”及“太热”。通常按偏差划分，常见的分为 7 个子集，即，负大偏差（NL）、负中偏差（NM）、负小偏差（NS）、零偏差（ZR）、正小偏差（PS）、正中偏差（PM）、正大偏差（PL）。其次，分完子集，还要依经验，确定各个模糊子集的隶属函数。

除了对被控量的值划分模糊子集，如需要，也可对被控量值的变化划分模糊子集，如上例，可划分水温正很快上升、正上升、现不变、正下降、正很快下降等子集，并建立相应的隶属函数。这种不仅检测被控量的值，而且检测被控量值的变化，并用这两者作为控制输入。这样的模糊控制就不是一维的，而是二维的模糊控制了。

如果把被控量值变化的变化，即被控量的“加速度”，也作上述处理，也作为模糊控制输入，那就是三维的模糊控制了。这样的系统当然要复杂些，程序量也大些，但控制的效果肯定要好些。

显然，这里的“划分模糊集”与“建立隶属函数”都要靠经验。没有经验，这个“划

分”与“建立函数”是难以实现的。

有的被控系统的被控量不只一个，而是两个或多个。则应把各被控量分别作相应的模糊化，都作为模糊控制输入。这样的模糊控制系统就不是单变量的，而是多变量的了。

不管是一维、二维或三维，还是单变量或多变量，其模糊化总是，先划分模糊子集，然后再建立其隶属函数。实现的方法都是用以上算法，设计 PLC 程序或使用 PLC 专用模块，靠 PLC 运行程序，把检测到的被控量的值转换为相应模糊子集的隶属度。

2. 模糊推理算法

模糊推理的任务是，根据当前输入的不同隶属度的模糊子集，遵照预先设定的规则，推断应有的模糊控制输出。这些也是靠运行 PLC 程序，或用 PLC 专用模块实现。

模糊推理依据是规则。常见的规则大体有：

“如 A 则 B”型，可写成：

IF A THEN B（如果 A 成立，那么 B 成立）

“如 A 则 B 否则 C”型，可写成：

IF A THEN B ELSE C（如果 A 成立，那么 B 成立；否则 C 成立）

“如 A 且 B 则 C”型，可写成：

IF A AND B THEN C（如果 A 成立 ，同时 B 也成立；那么 C 成立）

“如 A 或 B 则 C”型，可写成：

IF A OR B THEN C（如果 A 成立 ，或如果 B 也成立；那么 C 成立）

等等。

注意，这里的 A、B、C 都是模糊量的集合。如果小前提为 A1，与大前提 A 不完全一样，是否能推断出有价值的结论？这在传统逻辑推理中是不可能的。而用模糊逻辑推理则是可能的。如下显示的即为它的推理过程：

大前提	$A \to B$
小前提	A1
结论	$B1 = A1 \circ (A \to B)$

这里符号“∘”为矩阵乘。

以下还是以洗澡水温度控制为例，看看，有了“如 A 则 B”这个大前提后，是怎么进行模糊推理的。

这个大前提就是，水的状态（如“太热”等模糊子集）与控制输出（如“ZLL”等子集）间的关系 R。以上已介绍过，这是凭经验得出的。这个关系可用以下矩阵表达。这里“0”、“1”代表的是关系隶属度。

	ZLL	ZL	ZO	ZR	ZRR
太热	1	0	0	0	0
热	0	1	0	0	0
暖和	0	0	1	0	0
凉	0	0	0	1	0
太凉	0	0	0	0	1

小前提 A1 如为“太热”，即：

$$A1 = | 1\ 0\ 0\ 0\ 0 |$$

则 B1 = A1∘R。即

$$B1 = | 1\ 0\ 0\ 0\ 0 | \circ \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

按矩阵运算规则知，这时 B1 = | 1 0 0 0 0 | 。即仅“ZLL”子集的隶属度为 1，其它的均为 0。

小前提 A1 如不正好是“太热”，而是在“太热”与“热”之间，如水温为 70℃即：

A1 = | 0.5　0.5　0　0　0 |，则

$$B1 = | 0.5\quad 0.5\quad 0\quad 0\quad 0 | \circ \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

按矩阵运算规则知，这时 B1 = | 0.5　0.5　0　0　0 |。即“ZLL”子集的隶属度为 0.5，“ZL”子集的隶属度液为 0.5，其它的均为 0。

对“如 A 则 B 否则 C”型的推理，其“IF A THEN B”部分，与上相同。“ELSE C”部分，要用到模糊非隶属度的运算。这个运算较简单，即：

A 非的隶属度 = 1 − A 的隶属度

A1 非的隶属度 = 1 − A1 的隶属度

有了这个关系，其推理过程可描述如下：

大前提	$\overline{A} \rightarrow C$
小前提	$\overline{A1}$
结论	$C1 = \overline{A1} \circ (\overline{A} \rightarrow C)$

这里符号“∘”也是矩阵乘。再往下的处理与上类似，就不再赘述了。

对“如 A 且 B 则 C”型及“如 A 或 B 则 C”型的推理，如弄清模糊逻辑“与”及模糊逻辑“或”的算法，进一步处理也就不难了。

模糊逻辑“与”，其符号为“∧”，如：D = A ∧ B，则 D 的隶属度为 A、B 小者，即按最小原则定其隶属度。

模糊逻辑“或”，其符号为“∨”，如：D = A ∨ B，则 D 的隶属度为 A、B 大者，即按最大原则定其隶属度。

有了这些关系，其推理的进一步处理也与上类似，也就不再赘述了。

从以上介绍可知：

（1）模糊推理的算法是确定的，因此，总可用 PLC 相应程序予以实现。

（2）有了这个推理方法，同一大前提，但小前提不同，也可推出相应的不同结论。显然，有了它可减少规则数。

（3）推理的结论仍是模糊子集，要用于输出，还要解模糊处理。

提示：有的系统输入输出关系不是线性的。有的还有时滞，甚至是大滞后，有的还是不确定的，或无法建模的，等等。这对基于从经验总结出的语言型规则，去实现进行控制的模糊控制来说，是无关紧要的。完全可建立相应的控制策略予以实现，并可得到应有的效果。

3. 解模糊算法

模糊推理的输出仍是模糊量，是输出模糊子集。要用它用作控制输出，必须把这个输出模糊子集，按照一定算法转换为确定量的控制输出。这也就是解模糊。

为此，首先，要确定控制输出的类型。有两种控制输出：比例输出，积分输出及这二者相结合输出。

比例输出是把解模糊求得的值，直接用作控制输出。即它响应快，但为有静差控制。

积分输出是把解模糊求得的值，与当前控制输出先进行代数和，然后再用这个和作控制输出。它为无静差控制，但响应慢，且有可能超调或振荡。

也把这二者相结合，在一定情况下用比例输出，在另一情况下用积分输出。这样，有时可取得更好的效果。

其次，还要确定解模糊方法。有多种方法：如最大隶属度法、中位数法、加权平均法、估值法、重心法及求和法等。较常用的为最大隶属度法、加权平均法。

最大隶属度法较简单，选择出哪个隶属度最大，就用与其对应确定值用作输出。加权平均法就是按隶属度加权平均，把这样求得的值用作输出。

总之，解模糊实际是一些数据处理问题。这对具有较多运算指令的 PLC 处理起来是不难的。

4.9.3 模糊算法实现

本小节介绍的用 PLC 程序实现模糊控制算法，实际是上一节模糊控制算法介绍的注解。所用的实例也是洗澡水温度的模糊控制。整个程序由 3 部分组成：即模糊化程序、模糊推理程序、解模糊程序。

1. 模糊化程序

图 4-94 所示为模糊化子程序。它用于一个“Δ”或半“Δ”型的模糊子集的隶属度计算。程序所用的数据用符号表示，较好理解。

图 4-95 为调用它的主程序。程序所用的数据也用符号表示，也较好理解。但在每次调用它之前，先对“标志值形参”赋值，即指明该模糊子集隶属度为 1 时对应的精确量值，如“暖和”隶属度为 1 时，其是 40℃等；调用后，还得取出返回值，这里即为某输入模糊子集，如暖和等。

图 4-94 程序，先是求出“输入现值”与“标志值形参”（其实际值由计算哪个模糊子集确定）的差（差存于 DM72 中），并相应判断这两个值哪个大（用 LR0.00 表达，0 时前者大，1 时后者大）。然后，把这个“差”乘 10（积还存于 DM72 中）。再者，此“差”被“上半范围”或“下半范围”除（有可能“Δ”型不对称），其“商”仍存于 DM72 中。最后，用这个“商”去减 10。其结果存于“入模糊集形参”中。这也就是所求的隶属度。只是，它不是处于 0 ~ 1 之间的数，而是 0 ~ 10 之间的数。因为，这里用的都是整数运算指令，故作了这么处理了。

图 4-95 所示程序有 5 次调子程序。因为本例有 5 个模糊子集。

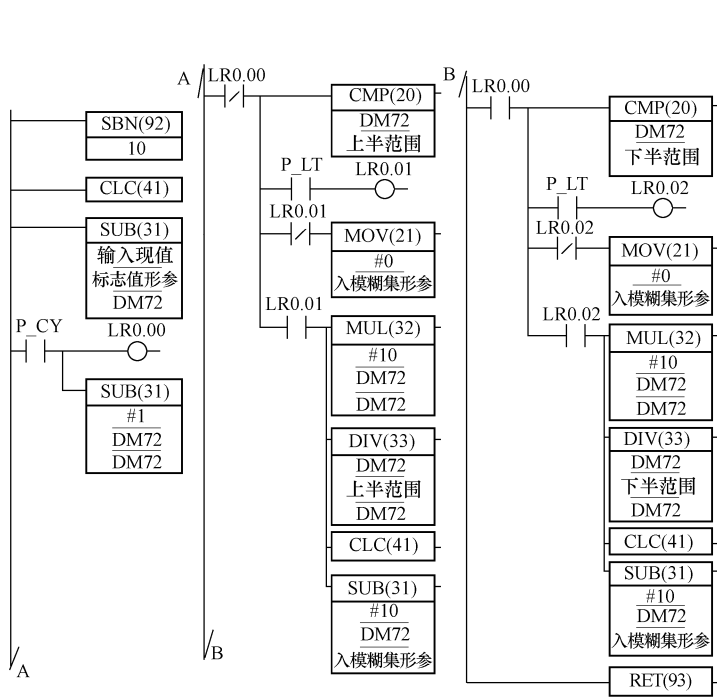


图 4-94　模糊化子程序

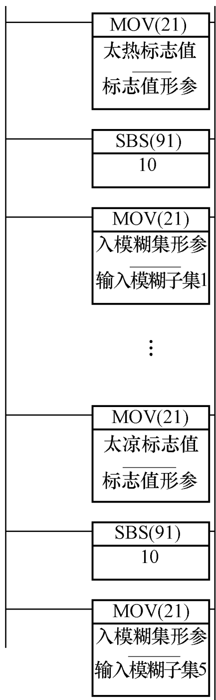


图 4-95　模糊化子程序调用

2. 模糊推理程序

图 4-96 所示为模糊推理子程序（子程序 11）及被它调用的下一层的子程序（子程序 12）。它用于进行一个行（本例为 5 行）与一个列（必须与行相等，本例也为 5 列）的矩阵乘。程序所用的数据用符号表示，较好理解。

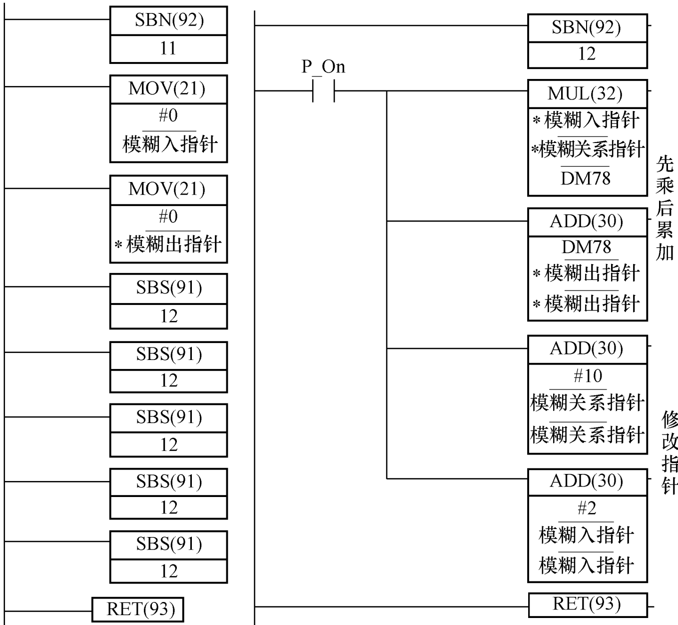


图 4-96　模糊推理子程序

图 4-97 所示为模糊推理子程序调用。程序所用的数据用符号表示。它在每次调用它之前，先对“模糊关系指针”及“模糊输出指针”赋值，即指明该模糊关系矩阵列的首地址及模糊输出地址。这种调子程序前赋值给指针，由子程序按指针指向去存储子程序运算结果，可作到一个子程序多用，是较简洁的编程方法。这里调了 5 次子程序，因为，这个矩阵有 5 列。

图 4-96 子程序 12，为先乘运算，后累加。运算后，为修改指针。子程序 11 先是模糊入指针赋值，指出输入矩阵的首地址；后为模糊出指针指向的地址清零，为累加作准备。以下为连续调 5 次子程序 12，以完成一个行与一个列的巨阵相乘。

显然，执行这主程序及对应的子程序，将完成整个推理过程。之后，可得出模糊输出矩阵。从图 4-97 子程序指针赋值知，此矩阵的各项的地址分别为：DM60 、DM62 、DM64、DM66 及 DM68。不过，这模糊输出子集的隶属度也是在 0 ~ 10 之间取值。

3. 解模糊程序

上已提及，解模糊程序的任务是，根据各输出模糊子集的隶属度，确定相应的控制输出。本例用的为积分输出。解模糊用的为最大隶属度法。图 4-98 及图 4-99 所示即为这个程序。该图指令操作数为符号地址。这里的 ZLL 对应的地址为 DM60（是模糊输出指针的指向，如图 4-97 所示，下同），ZL 对应的地址为 DM62，ZO 对应的地址为 DM64，ZR 对应的地址为 DM66，ZRR 对应的地址为 DM68。

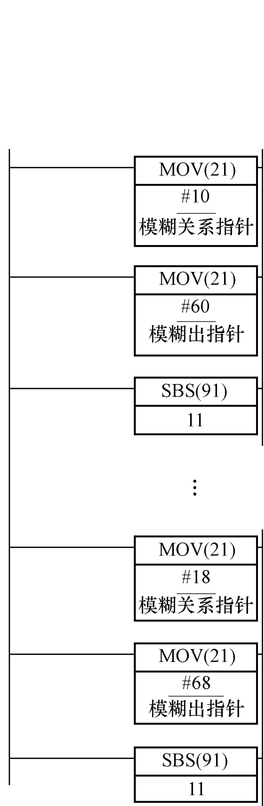


图 4-97 模糊推理子程序调用

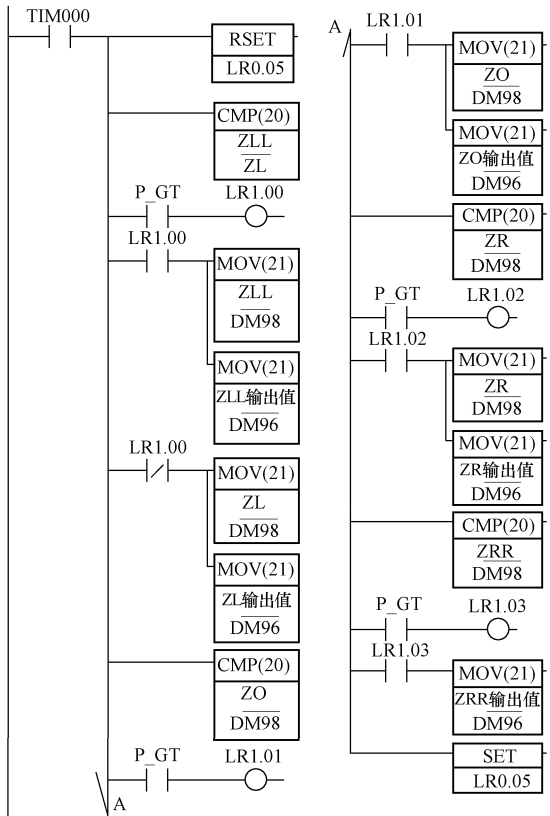


图 4-98 解模糊程序

图 4-98 程序用以查找最大隶属度的模糊子集。它用了多个比较指令，目的是把最大隶属度模糊子集的地址保存在 DM98 中。同时把这个模糊子集对应的输出值存于 DM96 中。为取得相应的控制输出准备数据。

图 4-99 程序是用以产生积分输出的。积分是使控制输出增大还是将减小，取决于 LR0.05 是 ON，还是 OFF。而 LR0.05 的 ON，OFF 则由那个模糊子集的隶属度大决定。在查找最大隶属度模糊子集的程序（图 4-98）的第一操作就是使 LR0.05 OFF，但如出现 ZR 或 ZRR 的隶属度最大（要使控制输出增大），则置 LR0.05 ON。如未出现 ZR 或 ZRR 的隶属度最大（要使控制输出减小），则保持 LR0.05 OFF。这样做正是系统所要求的控制。

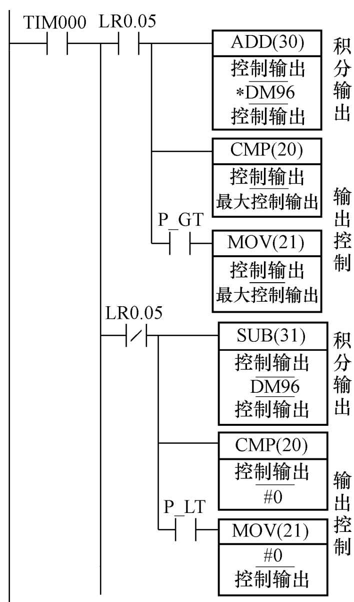


图 4-99　解模糊程序续

此外，还有两点要考虑：

（1）这个程序应是“微分执行”，其执行的时间间隔由 TIM000 控制。这个间隔可小些，但模糊子集对应的输出值也要小些。因为每次增减量小，可避免“静态不稳定”。

（2）控制输出值要控制，最大不能大过允许的的最大值；最小不能小于允许的的最小值。这在图 4-99 的程序中已作了处理。

4.9.4　模糊控制模块

为适应模拟量模糊控制的需要，OMRON 公司于 20 世纪末，开发了分别用于大、中型机的模糊控制单元。图 4-100所示为 C200H-FZ001 模糊控制单元外观。

它可处理 8 个输入、4 个输出。最多有可建立 128 个规则。每个规则可有 8 个条件与两个结论。

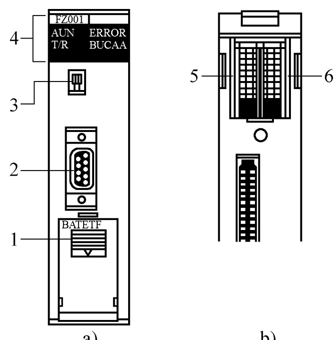


图 4-100　模糊模块
a）面板　b）背板
1—内装电池　2—RS-232 通信口　3—单元号开关　4—指示灯　5—DIP 开关 1　6—DIP 开关 2

1. 内存区分配

用机号设定开关指定机号 n（n = 0 ~ 9，不能与其它特殊古单元的机号重复）后，有 5（100 + 10n 到 100 + 10n + n-1）个字为本单元专用。这些字的功能见表 4-20 ~ 表 4-22。

2. 使用

以下以控制传送带 B 速度的实例，如图 4-101 所示，说明这个单元的使用。从图知，这里有两个传送带，带 A 用以输送产品，等速运动，但产品间的间隔是随机的；带 B 用以输送包装盒，盒的间隔是相等的，但速度是要调整的。同时，有 4 个光电开关，PH1、PH2、PH3 及 PH4，用以检测产品（PH1、PH3）或包装盒（PH2、PH4）通过的情况。还有两个旋转编码器 A、B，分别安装在 A、B 传送带的驱动电动机轴上（未示出），用以检测产品与包装盒间的位置偏差。

表 4-20　字的功能表

IR 字	位	功　　能
1n0	00 ~ 03	输入数(1 ~ 8,BCD)
	04 ~ 14	无用
	15	工作开始位
1n1	00 ~ 15	第一个输入字
1n2	00 ~ 03	输出数(1 ~ 4,BCD)
	04 ~ 15	无用
1n3	00 ~ 15	第一个输出字
1n4	00 ~ 15	单元标志字

表 4-21　字的功能表（续 1）

数据区	位 15	位 14	位 13	位 12	IR 1n1 最高数位值
DM	0	0	0	0	0
IR or SR	1	0	0	0	8
HR	0	1	0	0	4
AR	1	1	0	0	C
LR	0	0	1	0	2
TC	1	0	1	0	A

表 4-22　字的功能表（续 2）

IR 1n4 位	标　志　名　称	功　　能
00	结果输出标志	ON,说明模糊单元经处理已产生输出
01 和 02	…	无用
03	设定错标志	ON,说明设定错
04	单元出错标志	ON,说明单元自诊断错
05	内存出错标志	ON,说明知识库出错
06	电池电压不足	ON,说明电池电压不足或单元未安装电池
07	处理允许标志	ON,说明单元可进行模糊处理
08 ~ 15	…	无用

当产品通过 PH1 时，检测产品的 PLC 的高速计数单元记录旋转编码器输入的脉冲，以计算产品离开 PH1 处的距离。而当包装盒通过 PH2 时，检测包装盒的 PLC 的高速计数单元记录旋转编码器输入的脉冲，以计算包装盒离开 PH2 处的距离。这两个距离差，也就是两者的位置偏差。当产品到了 PH3 位置，而包装盒又正好到了 PH4 的位置时，传送带上的机械手把产品推向包装盒。产品装入包装盒后，将在传送带 B 的带动下，送到目的地。

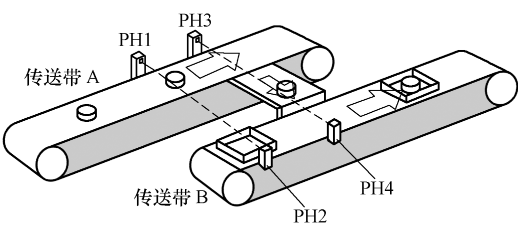


图 4-101　传送带 B 速度控制

由于产品的间距常是不相等的，为了确保产品到达 PH3 时，正好包装盒也正好到达 PH4，须经常调整传送带 B 的速度，以保证产品与包装盒位置偏差及其变化率合乎模糊控制要求。

这里用到的第 1 个模糊规则就是：如果产品包装盒位置偏差大，那么，传送带 B 速度加快；如果产品包装盒位置偏差小，那么，传送带 B 速度不必太加快。当然，这里假设每次只有一个产品通过。

令 E 代表偏差，则有：

E = 旋转编码器 A 计数值 − 旋转编码器 B 计数值

此外，还有输入的数据为偏差 E 的变化 ΔE：：

ΔE：=（En）−（En-1）

3. 输入变量模糊化

设 5 个模糊集合，反映这个偏差。这 5 个集合分别是负大（NL）、负小（NM）、零偏差（ZR）、正小（PM）、正大（PL）。各模糊集合的隶属度与偏差的关系也都是线性的，分别为“Δ”，或半“Δ”型。如图 4-102 所示。

也设 5 个模糊集合，反映这个偏差变化。这 5 个集合分别是负大（NL）、负小（NM）、零偏差（ZR）、正小（PM）、正大（PL）。各模糊集合的隶属度与偏差变化的关系也都是线性的，分别为“Δ”，或半“Δ”型。如图 4-103 所示。

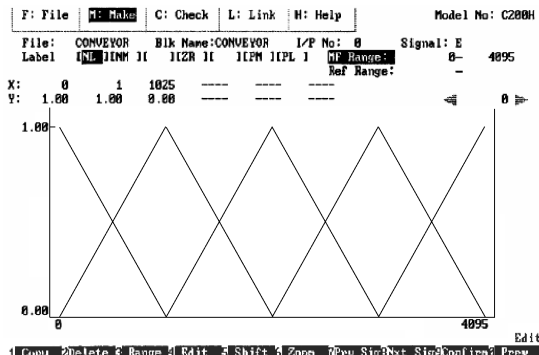


图 4-102　偏差模糊集合隶属度

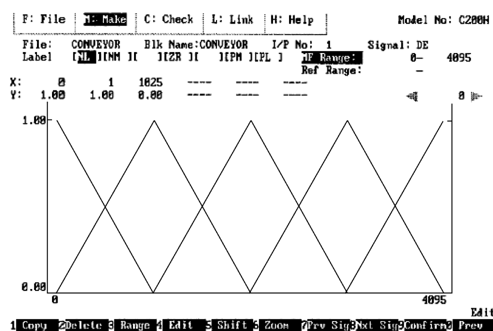


图 4-103　偏差变化模糊集合隶属度

4. 输出量的解模糊

也设 7 个模糊集合，反映这个偏差变化。这 5 个集合分别是负大（NL）、负中（NM）、负小（NS）、零偏差（ZR）、正小（PS）、正中（PM）、正大（PL）。各模糊集合的隶属度与偏差变化的关系也都是线性的，分别为直线型。如图 4-104 所示。

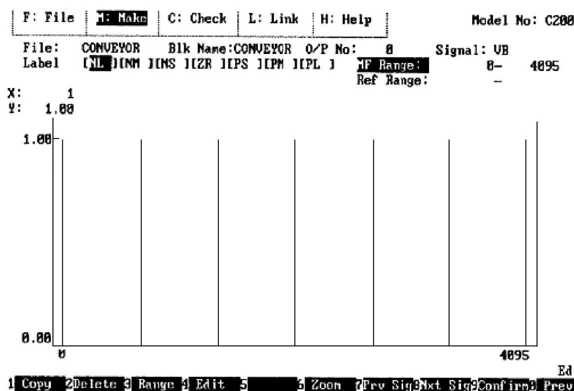


图 4-104　输出模糊集合隶属度

这三组集合与图 4-92 不同的是，它用 OMRON 提供模糊单元有关软件，FSS 软件建立的。FSS 软件界面友好，

用下拉式菜单进行操作，即使计算机的初学者，只要对系统熟悉，按提出要求作，也不感困难。

5. 建立规则

规则是依实际经或专家验建立的。当 E、ΔE 处于不同情况时，依经验对应的输出见表 4-23。

表 4-23 输出表

ΔE \ E	包装盒超前	包装盒稍超前	包装盒正好	包装盒稍落后	包装盒落后
包装盒速度快	包装盒大减速	包装盒大减速	包装盒稍减速	包装盒稍加速	包装盒加速
包装盒速度稍快	包装盒大减速	包装盒减速	包装盒稍减速	包装盒稍加速	包装盒加速
包装盒速度正好	包装盒减速	包装盒稍减速	包装盒速度不变	包装盒稍加速	包装盒加速
包装盒速度稍慢	包装盒减速	包装盒稍减速	包装盒稍加速	包装盒加速	包装盒大减速
包装盒速度慢	包装盒减速	包装盒稍减速	包装盒稍加速	包装盒加速	包装盒大加速

若用符号表示，则应为表 4-24。这里 DE 即为 ΔE。NL 即为包装盒大减速、包装盒速度慢、装盒超前。NM 即为包装盒减速。NS 即为包装盒稍减速、包装盒速度稍慢、包装盒稍超前。等等。

表 4-24 符号表达的输出表

DE \ E	NL	NS	ZR	PS	PL
PL	NL	NL	NS	PS	PM
PS	NL	NM	NS	PS	PM
ZR	NM	NS	ZR	PS	PM
NS	NM	NS	PS	PM	PL
NL	NM	NS	PS	PM	PL

如果将表 4-24 的含义转换为规则，则为

IF E = NL AND DE = PL THEN VB = NL

IF E = NL AND DE = PS THEN VB = NL

IF E = NL AND DE = ZR THEN VB = NM

IF E = NL AND DE = NS THEN VB = NM

IF E = NL AND DE = NL THEN VB = NM

IF E = NS AND DE = PL THEN VB = NI

IF E = NS AND DE = PS THEN VB = NM

IF E = NS AND DE = ZR THEN VB = NS

IF E = NS AND DE = NS THEN VB = NS

IF E = NS AND DE = NL THEN VB = NS

IF E = ZR AND DE = PL THEN VB = NS

IF E = ZR AND DE = PS THEN VB = NS

IF E = ZR AND DE = ZR THEN VB = ZR

IF E = ZR AND DE = NS THEN VB = PS
IF E = ZR AND DE = NL THEN VB = PS
IF E = PS AND DE = PL THEN VB = PS
IF E = PS AND DE = PS THEN VB = PS
IF E = PS AND DE = ZR THEN VB = PS
IF E = PS AND DE = NS THEN VB = PM
IF E = PS AND DE = NL THEN VB = PM
IF E = PL AND DE = PL THEN VB = PM
IF E = PL AND DE = PS THEN VB = PM
IF E = PL AND DE = ZR THEN VB = PM
IF E = PL AND DE = NS THEN VB = PL
IF E = PL AND DE = NL THEN VB = PL

当然，这些规则须下给模糊控制模块。只有这样，模块才可，依照这些规则实施控制。

6. I/O 分配

本例各输入、输出及特殊单元的机号及地址分配见表 4-25。

表 4-25　机号及地址分配

模　　块	模块(机)号	I/O 字
开关量输入单元	…	IR000
开关量输出单元	…	IR001
高速计数单元 A	0	IR100 ~ IR109
高速计数单元 B	1	IR110 ~ IR119
模入单元	2	IR120 ~ IR129
模出单元	3	IR130 ~ IR139
模糊控制单元	4	IR140 ~ IR149

模入单元的用以检测两个传送带驱动电动机的转速。121 通道反映 A 带电动机转速，123 通道反映 B 带电动机转速。

模出单元用以控制带 B 的驱动电动机。用其 130 通道的值，控制其转速。

本例各输入、输出开关量地址分配见表 4-26。

表 4-26　输入、输出开关量地址分配

模　　块	I/O 位	功　　能
输入单元	00002	PH1
	00003	PH3
	00005	PH1
	00007	PH4
输出单元	00102	产品推向包装盒
	00104	传送带 A 工作
	00105	传送带 B 工作

7. DM 区设定

高速计数单元的有关设定见表 4-27。

表 4-27　高速计数单元设定

字	值	功　能
DM1000	0400	指定高速计数器 A 为门控模式
DM1001	0001	指定高速计数器 A 的计数开始命令位为 10003
DM1100	0400	指定高速计数器 B 为门控模式
DM1101	0001	指定高速计数器 B 的计数开始命令位为 11003

模入、模出单元也要按要求，作相应的设定，具体略。

8. 梯形图程序

（1）初始化。进行有关模块的初始化设定。控制逻辑：进行起动、停车及有关控制。具体略。

（2）输入处理。采集包装盒与产品的位置偏差及偏差变化，并按间隔设定系数进行转换，然后送模糊控制单元的输入 1 及输入 2。图 4-105 所示为采集包装盒与产品的位移计数程序。

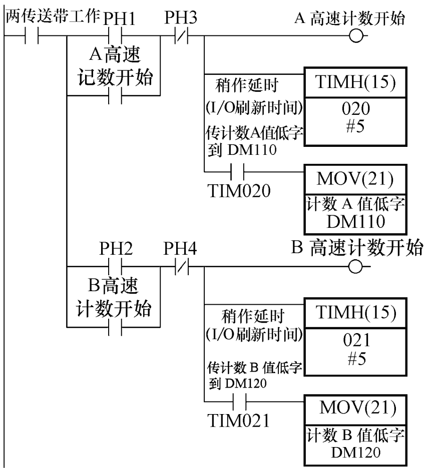


图 4-105　输入处理程序

从图知，当 PH1 ON 及 PH3 OFF ，即产品到达 PH1，而又未到达 PH3，则发出 A 高速计数开始命令。稍作延时（命令到执行有个 I/O 刷新的过程），后则不停地把计数器 A 的现值送 DM110。当 PH2 ON 及 PH4 OFF ，即包装盒到达 PH2，而又未到达 PH4，则发出 B 高速计数开始命令。稍作延时（命令到执行有个 I/O 刷新的过程），后则不停地把计数器 A 的现值送 DM120。

产品到达 PH3 时，PH3 ON A 计数停止，并复位。包装盒到达 PH4 时，PH4 ON B 计数停止，并复位。同时，DM110、DM120 复位、置 0（如图 4-106 中梯形图程序）。

图 4-106 所示为包装盒与产品的位置偏差及偏差变化计算、按间隔设定系数进行转换及送模糊控制单元的输入 1 及输入 2 的程序。

从图知，先是当前偏差复制，然后为计算偏差。接着，计算偏差变化。这时，如产品到达 PH3、或包装盒到达 PH4，则 DM110、DM120 复位。

为了，可调整模糊化域的间距，这里可设偏差系数、偏差变化系数，并用这系数乘偏差、偏差变化，以得到偏差修正值及偏差变化修正值。

进而，求偏差及其变化的绝对值。这么做是为了与图 4-102 及图 4-103 设置相一致。最后，再对偏差及其变化的绝对值进行控制，以避免超出模糊单元的输入范围。作了以上处理后，则把偏差及其变化这两个量作为模糊单元的第一及第二输入送模糊单元。只要模糊单元

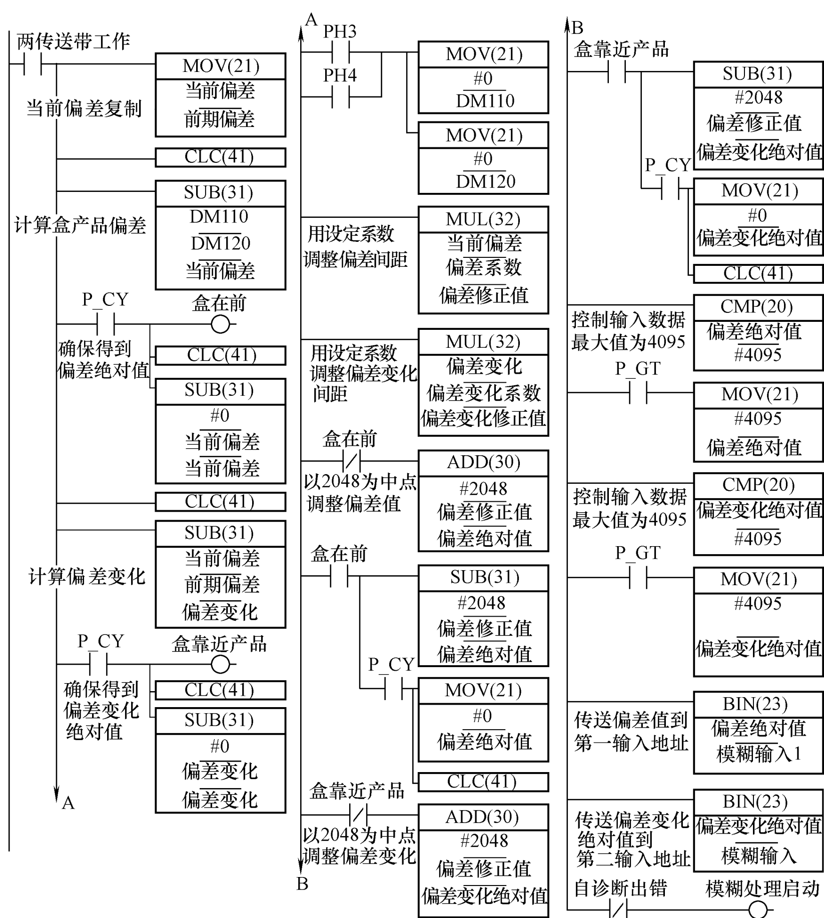


图 4-106　位置偏差及偏差变化计算程序

设定正确、工作正常，则起动模糊处理。

给了模糊单元输入，起动模糊处理后，模糊控制单元会按下载的规则，进行模糊化、模糊推理，并产生相应输出。这些都无须用户干预与编程。这是使用模糊控制模块的最大优点。

（3）输出处理。把模糊控制处理后产生的输出，用于控制传送带 B 的电动机，还须作相应处理。有两种处理方法：一为绝对输出处理，另一为相对输出处理。

1）绝对输出处理。它把模糊单元的输出，直接转换为控制传送带 B 电动机的模出通道 130。这么作，系统是有静差的。

2）相对输出处理。它把模糊单元的输出，先转换为相对值，再将这相对值与 130 的现值作代数和。这么作，系统是有无静差的。图 4-107 所示即为这个程序。

从图知，这里，在输出前，还对输出的最大值及最小值作了控制。使其不大过 4095（十六进制为 FFF，为模出单元的最大值），不小过 DM185（此值为带 A 电动机转速的 1/10）。

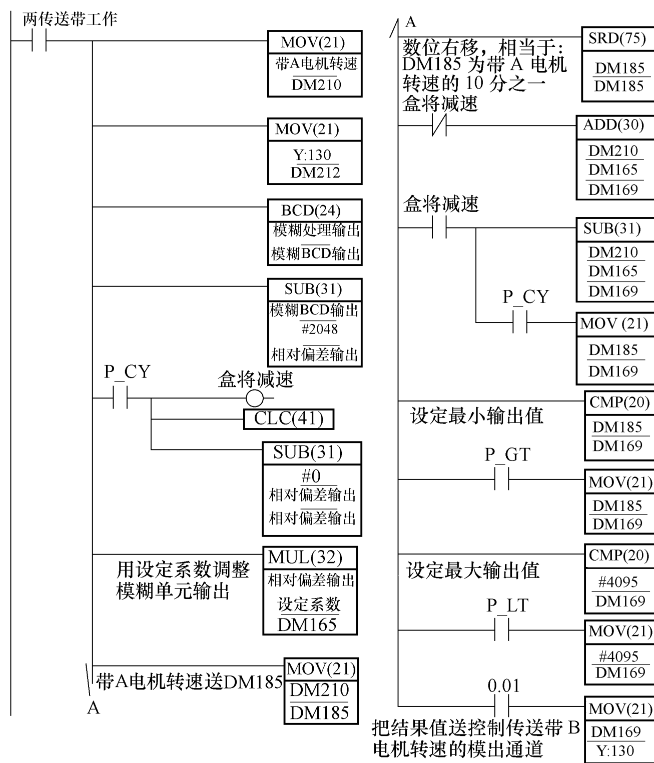


图 4-107 相对输出处理程序

4.10 智能控制

关键词：最优控制、自适应控制、鲁棒控制、自学习控制、预测控制、专家控制、复合控制

随着现代控制及 PLC 技术的飞速发展，高级控制在 PLC 的模拟量控制中，也得到越来越多的应用。这些高级控制有：最优控制，自适应控制，自学习控制，预测控制，专家控制及复合控制等等。

越来越多的实践说明，这些控制对具有多变量、非线性、时变性和不确定性，难于建立精确数学模型的实际工业过程或复杂系统比传统控制更为有效。

前已指出，“模拟量控制程序设计，与其说是取决于设计者对 PLC 的了解，不如说是取决于设计者对自控知识的掌握；它的难点，似乎不在于 PLC 程序本身，而在于要很好地运用好有关自控知识”。特别在本节涉及的各种控制中，将更是如此。

所以，本章的讨论将只集中在原理的简单介绍上。具体的程序只好由读者，依此设想，按需要与可能去设计了。

4.10.1 最优控制

最优控制（Optimal Control）是指，怎样选择控制规律使控制系统的性能和品质在某种意义下为最优。如最小时间控制，最少燃料控制和最佳调节器等。

最优控制已经在航天、航海、导弹、电力系统、控制装置、生产设备和生产过程中得到了比较成功的

应用，而且在经济系统和社会系统中也得到了广泛的应用。

1. 最优控制发展

在第二次世界大战期间及其后的一段时间内，以提高发射命中率为主要目标的伺服（随动）系统理论得到了迅猛的发展。这一理论在设计与分析单输入、单输出的，线性、时不变的集中参数系统时是行之有效的。然而，随着空间技术的发展，控制系统日趋复杂，其精度要求越来越高，这种理论就日益暴露出它的局限性来。因此，人们又开始寻找新的理论了。

早在 20 世纪 50 年代初期，就发表了从工程观点研究最短时间控制问题的文章，尽管其最优性的证明多半是借助于几何图形，带有启发的性质，但是毕竟为发展现代控制理论提供了第一批实际模型。

由于最优控制问题引人注目的严格表述形式，更因为空间技术的迫切需要，引起了一大批数学家的注意。人们发现，最优控制问题就其本质来说，乃是一个变分学问题。然而，经典的变分理论所能解决的，只是其容许控制属于开集的一类最优控制问题，而实际上遇到更多的，却是其容许控制属于闭集的一类最优控制问题，这就要求人们开辟求解最优控制问题的新途径。

在解决最优控制问题的种种方法中，有两种方法最富有成效，一种是美国学者贝尔曼（Bellman，R.）的“动态规划”；另一种是苏联学者庞特里雅金的“最大值原理”。

“动态规划”是贝尔曼在 1953 ~ 1957 年间逐步创立的，他依据最优性原理，发展了变分学中的哈密顿——雅可比理论，构成了“动态规划”。

“最大值原理”是庞特里雅金等人在 1956 ~ 1958 年间逐步创立的，他受到力学中哈密顿原理的启发，先是推测出“最大值原理”，随后又提供了一种证明方法，并于 1958 年在爱丁堡召开的国际数学会议上首次宣读。

最优控制的一个前提是，系统的控制是否存在最优解？而这个判定是比较复杂的。故一般都是先假定有一个最优解，然后去求这个最优解。最优解一般总是代表系统工作状态的极值点，所以，有时称其为极值控制系统。

还应了解，所求得最优应是“整体”最优，而不是局部最优。一般很难用定量方法求得整体最优。因此，常常是求出许多局部最优，再挑选整体最优。

再就是要了解，怎么去自搜寻最优，并使系统保持在最优点附近的工作。

2. 最优控制算法

最优控制关键在自寻最优。所以，它的算法主要是怎样去自寻最优。

PLC 模拟量最优控制是基于计算机的控制，是通过运行程序实现的。其寻求最优的过程不必弄清系统的数学模型，而是靠实际试探。由于当今的 PLC 已经有很强的运算能力，很大的内存容量以及很快的指令执行速度，已完全可以进行这个探索。

最优控制的过程如图 4-108 所示。

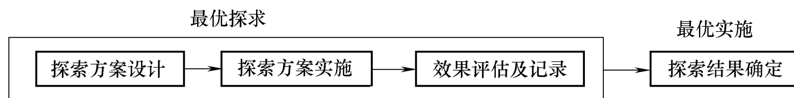


图 4-108　最优控制过程

可知，自寻最优系统应具有两个基本功能：①实时地不断检测本身的工作状态，不断地对系统是否处于当时可能达到的最优状态作出判断。②根据检测和判断所得的信息迅速地作出使系统趋向最优状态的调整。

实现自寻最优点的方法主要有切换、摄动、自导和模型定向等。

（1）切换法。先让输入信号以恒定速度沿着促使系统性能改善的方向变化，直到系统性能不再继续改善时为止。这时系统状态可能已越过最优点，但仍处在最优点附近。然后，再让输入信号以同样速度反向变化，重复同样的调整过程。不断地重复这种调整，就可能使系统的工作状态保持在最优点附近。

输入信号作用方向的变化周期称为搜索周期，自寻最优系统的性能值只能在最优点附近变化而不能严格保持在最优值，这种偏差的平均值称为搜索损失。

较长的搜索周期可使系统具有较强的抗干扰能力，但同时也加大了搜索损失。设计自寻最优系统时，应结合实际情况兼顾各方面的性能要求而选取适当的搜索周期。

（2）摄动法。这种方法是切换法的发展。其基本原理是：在系统的输入量上，附加一个周期性的探测信号（摄动量），例如正弦信号。通过在这个混合输入信号作用下，系统输出信号的分析，可以得到系统输出是否向最优点运动的信息。利用这个信息不断地调整输入量即可使系统向最优点运动。采用这种搜索方式时，可根据噪声特性、系统动态特性和对搜索损失的要求来确定探测信号的幅度和周期，以及是否需要采取移相措施等。

（3）自导法。这种方法不需要其它输入信号。将输出量对时间的一阶导数的实测值，经积分作用后送入系统的输入端，以实现对极值点的搜索。取积分的目的是为了减弱或消除外界或内部随机干扰的影响。只要开始时输出量的变动趋势与寻优方向相符，系统就会自动趋向于最优点。具体设计这种系统时，必须考虑系统的动态特性对最优点的影响。

（4）模型定向法。基本思路是建立一个能描述最优运动状态的数学模型（系统的最优点可由此模型确定），再根据这个模型用一定的算法找出所需的输入。为建立这个模型，可以先向系统输入一个试验信号，然后根据系统的输出数据用参数估计方法确定模型的各个参数。模型定向法需要较多的数值计算，用于在线控制时需要使用快速计算机。

具体的程序可分块实现。这些程序块，有：效果评定程序模块，试探计划程序模块，试探执行程序模块，方案确定程序模块等。具体程序略。

时至今日，最优控制理论的研究，无论在深度或是广度上，都有了较大的进展。然而，随着人们对客观世界认识的不断深化，又提出了一系列有待解决的新问题。可以毫不夸张地说，最优控制理论依旧是极其活跃的科学领域之一，也是 PLC 实现模拟量控制的重要方面。

3. 最优控制应用

最优控制理论问题较为深奥，但 PLC 模拟量用的最优控制却没那么复杂。只是有这么几种情况：

在控制算法确定后，怎样选定控制参数，使控制效果最佳。如使用 PID 控制时，其控制参数自行整定，也可说成这里的最优控制。而且是自寻最优控制的。再如，在比例控制中，比例系数选定，也有个怎么选择，以实现最优的问题。

在控制结构确定之后，一般都有多种控制算法，如用 PID，或只用 PD，而不用 I，也可能用模糊控制算法等等。怎样在其中自动选定一个最优的算法。

控制结构，即用那些控制量，通过那些环节实施对系统的控制，如有多种方案可选定，那也可设计相关的最优控制算法，以确保被控制量的变化，或系统的投入、产出获得最佳。

实施这些最优算法，关键要处理好两个问题：

一是，“优”的评价。什么叫“优”？要有标准。

二是，可选用方案的设计。根据经验，一般讲，最优方案总是在可预料的范围内出现。即实际的系统，最优可选变量的定义域总是有边界的。这样，即可在这个定义域内设计可能的方案。

有了这两条，再加上相应搜索程序。实现最优控制是可能的。

4.10.2 自适应控制

1. 自适应控制（Adaptive Control）概念

适应原为生物学的术语。指的是生物能改变自己的习性以适应新的环境的一种特征。自适应控制是指这样的控制，它能不断地检测系统的信息，自动调整控制的参数以至算法、结构，以适应环境条件或过程参数的变化，以使系统始终具有较高的控制性能。

适应控制的基础是系统在线辨识及自身调整。即在系统的运行过程中，依据对象的输入、输出数据，不断地辨识系统。使控制变得越来越准确，越来越接近于实际。

如当系统在设计阶段，由于对象特性的初始信息比较缺乏，系统在刚投入运行时可能不理想。但在实施控制的同时，进行在线辨识及自身调整，控制系统逐渐适应，最终将调整到一个满意的工作状态。再比如某些控制对象，其特性可能在运行过程中要发生较大的变化，但通过在线辨识和改变控制器参数，系统也能逐渐适应。

对那些对象特性或扰动特性变化范围很大，同时又要求经常保持高性能指标的一类系统，采取自适应控制是合适的。

图 4-109 所示为自适应控制系统的结构。

从图知，它比普通的反馈控制系统增加了一个适应控制回路。自适应控制器根据受控对象的输入、输出关系，辨识受控对象和外部干扰的特性。随后根据辨识的结果校正反馈控制规律，以适应环境特性的变化。无论辨识，还是控制规律的设计，都可采用不同的方法。它们的不同组合能形成适应控制的不同方案。

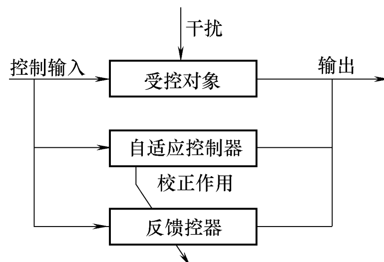


图 4-109　自适应控制系统结构

2. 适应控制特点

如果自动调节处理的是输入的干扰，而自适应控制处理的是整个环境的干扰。与常规反馈控制相比，有以下特点：（1）能辨识环境条件参数变化，建立被控过程的模型；（2）在辨识的基础上综合一种新的控制算法；（3）根据综合出的控制策略自动地修正控制器参数值。

适应控制发展：

适应控制系统主要类型有：自校正控制系统、模型参考适应控制和自寻最优控制系统。

利用实时辨识技术自动校正系统特性的适应控制系统。自校正调节器具有对系统或控制器参数进行在线估计的能力，可通过实时地识别系统和环境的变化来相应地自动修改参数，使闭环控制系统达到期望的性能指标或控制目标，有一定的适应性。

在经典控制理论和最优控制理论中，控制器的设计方法是建立在系统数学模型不变或事先已知的基础上的。但很多实际系统的数学模型是无法确切了解的，并且随着环境变化系统特性也在改变。因此，通常的非适应控制技术不能在线建立数学模型，也不能实时调整系统的参数。

为了克服通常的控制技术的这些缺点，R. 卡尔曼于 1958 年提出自校正调节器的设想。但由于当时适应控制理论尚未充分发展，又缺乏适用的计算机，卡尔曼的设想未能得到进一步的发展，更未能付诸实施。

1970 年，V. 彼特卡把自校正调节器的理论研究推广到随机情况。其后，随着随机控制理论、系统辨识理论和计算机技术的发展，自校正调节器的研究和应用迅速发展起来。

1973 年 K. J. 阿斯特勒姆和 B. 维滕马克提出一种简易可行的自校正调节器实现方案，引起广泛重视。在这个方案中，用一个表示输入输出关系的线性差分方程（可以包含干扰项）作为系统的预测数学模型（称为可控自回归滑动平均模型，缩写为 CARMA），用递推最小二乘法在线估计模型的参数，直接得到一个输出方差最小的自校正调节器（如图所示自校正调节器）。这种方案中系统的组成结构简单，容易实现，并易于在工业过程控制中推广。它的缺点是对于非最小相位系统控制过程可能发散。

1975 年，D. W. 克拉克和 P. J. 高思罗普又提出广义输出最小方差的自校正调节器方案，不但能限制控制输入的幅度，还能限制输出与设定值之间的误差，能同时用于最小相位系统和非最小相位系统。

1979 年 P. E. 韦尔斯泰德等人提出具有零极点配置功能的自校正调节器，能够在线整定系统或控制器的参数，使闭环系统的零点和极点配置到指定部位上去。随后，针对各种不同性质的系统（多变量系统、非线性系统、分布参数系统、时变系统和连续系统等）提出了相应的自校正调节器方案。此外，还出现了一些专用性的自校正调节器，如自校正 LQG（线性二次高斯）调节器，自校正 PID（比例积分微分）调节器等。

在工作原理上，自校正调节器是以分离原理为依据的，把参数的估计和控制律的计算分开进行。参数

估计采用递推方法，计算量较小，易于用计算机实现。自校正调节器已在不少工程技术领域（如造纸、化工、冶金、水泥、热力、船舶和飞机的自动驾驶装置、机械手等）中被采用，取得了较好效果。

适应控制系统主要用于过程模型未知或过程模型结构已知和参数未知且随机的系统中控制器参数的调整。

近年来，对适应控制的研究，大都建立在系统是逆稳定的基础上。而且在适应控制器中由Riccati方程确定的量。近期，有人通过对“一步超前”适应控制器的分析，提出“输入匹配”方法，将系统的信号跟踪问题转化为对系统输入的研究。还有人用随机梯度算法，建立了关于“输入匹配”全局收敛的“一步超前”最优适应控制器。又有人建立了随机系统“一步超前”最优适应控制的最小二乘算法，估计出由该算法辨识系统参数的收敛速率。适应控制系统的进一步发展，将走向“自学习”系统和“智能控制”系统。

3. 适应控制应用

适应控制在PLC控制中可设想的应用大体有，在控制算法上，如有几个算法可适用各种不同的“环境条件”。那么，系统工作时，在可检测反馈量的同时，还要检测这个“环境条件”，并依检测的“环境条件”的不同或变化，而选择控制的算法。以使系统的控制达到最好的效果。

不仅在算法上，其它的，如控制结构、控制参数，也都可准备多种方案，提供系统“环境条件”变化时选择，以使系统能“适应”“环境变化”，始终能在最佳的状态下工作。

4.10.3 预测控制

1. 预测控制（Predictive Control）产生

20世纪60年代初形成的现代控制理论，在航天领域取得了辉煌成果，利用状态空间法分析和设计系统，提高了对被控制对象的洞察能力，提供了在更高层次上设计控制系统的手段。但是，它在工业过程控制应用中，遇到了3大困难：

（1）没有或极难找到精确的数字模型。

（2）被控制对象的结构、参数和环境具有很大的不确定性。

（3）要求控制手段经济性。

这3大难题阻碍了现代控制理论在复杂工业过程中的有效应用。这个工业实践向控制理论的挑战促使预测控制，一种新型的基于计算机的控制的诞生。

预测控制采用多步测试、滚动优化和反馈校正等控制策略，因而控制效果好，适用于控制不易建立精确数字模型，且比较复杂的工业生产过程，所以它一出现就受到国内外工程界的重视，并已在石油、化工、电力、冶金、机械等工业部门的控制系统得到了成功的应用。

70年代，人们除了加强对生产过程的建模、系统辨识、自适应控制等方面的研究外，开始打破传统的控制思想的观念，试图面向工业开发出一种对各种模型要求低、在线计算方便、控制综合效果好的新型算法。

在这样的背景下，预测控制的一种，也就是模型算法控制（MAC-Model Algorithmic Control）首先在法国的工业控制中得到应用。因此，预测控制不是某一种统一理论的产物，而是工业实践中逐渐发展起来的。同时，计算机技术的发展也为算法的实现提供了物质基础。

现在比较流行的算法包括有：模型算法控制（MAC）；动态矩阵控制（DMC）；广义预测控制（GPC）；广义预测极点（GPP）控制；内模控制（IMC）；推理控制（IC）等等。

70年代在美、法等国的工业过程领域内出现的预测控制，在锅炉、分馏塔及石油加工生产装备上获得了成功应用。现已有成熟的商品软件包供应。我国在齐鲁石化首先引进了这一软件，应用也很成功。

2. 预测控制的基本思想

预测控制是一种基于模型的控制算法，这一模型称为预测模型。预测模型的功能是根据被控对象的历史信息和未来输入，预测其未来输出。

输入控制量为阶跃函数或脉冲函数。被控量（输出）的采样值为

$$a_i = a(iT) \quad i = 0,1,\Lambda,N$$

其中 T 为采样周期，当 $t = NT$ 时，认为系统的输出值已稳定。向量 $[a_0, a_1, \Lambda, a_N]^T$ 称为模型向量（非参数模型），N 称为建模时域。同时认为系统是线性的，满足叠加原理：

若 $$f(t) = y(t)$$

则 $$\alpha f(t) + \beta f(t) = (\alpha + \beta) f(t)$$

其意义是，可以根据所加控制量，利用模型向量计算出（预测）被控量（输出）。

预测控制是一种优化控制算法，通过性能指标的最优来确定未来的控制量。这一性能指标涉及系统未来的行为。

（1）根据实践经验，提出系统输出的曲线形式。

（2）确定一系列优化性能指标。

（3）根据优化性能指标求出 M 个未来时刻的控制量，只用下一时刻的控制量去实际控制。

预测控制是一种闭环控制算法。在通过优化确定了一系列未来的控制量后，为了防止模型失配或环境干扰引起控制对理想状态的偏离，预测控制通常不把这些优化计算出的控制量逐一全部实施，而只是实现本时刻的控制量。到下一时刻，首先检测被控对象的实际输出，利用这一实时信息对基于模型的预测进行修正，然后在进行新的优化。

反馈校正的形式多种多样，如可以把预测值和实际测量值的差（即预测误差）加权后与下一时刻的预测值的和作为下一时刻的真正预测值来优化计算控制量。使得优化不仅基于模型，而且利用了反馈信息，构成了闭环优化。

下面以模型算法控制为例子来说明预测控制的基本原理，如图 4-110 所示。

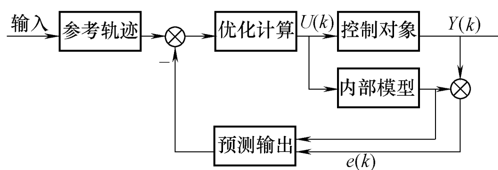


图 4-110　预测控制基本原理

模型算法（MAC）控制主要包括内部模型、反馈校正、滚动优化和参数输入轨迹等几个部分。它采用基于脉冲相应的非参数模型作为内部模型，用过去和未来的输入输出状态，根据内部模型，预测系统未来的输出状态。经过用模型输出误差进行反馈校正以后，再与参考轨迹进行比较，应用二次型性能指标进行滚动、优化，然后再计算当前时刻加于系统的控制，完成整个动作循环。

3. 预测控制应用

预测控制伴随着工业的发展而来，所以，预测控制与工业生产有着紧密的结合，火电厂钢球磨煤机是一个多变量、大滞后、强耦合的控制对象，其数学模型很难准确建立。而目前国内火电厂所装设的控制器大部分是 PID 控制器。由于系统各变量耦合严重，PID 控制器很难适应，致使钢球磨煤机不能投入自动运行。曾有人用 8051 单片机加上 A/D 8 路接口及其接口电路，再加上控制键和显示器，组成了预测控制器。在采用了 MAC 算法之后，就能够弥补 PID 控制器的不足。

8051 单片机加上 A/D 8 路接口及其接口电路能做到的，功能比起它要强大得多的 PLC 难道做不到吗？

预测控制具有适应复杂生产过程控制的特点，具有强大的生命力。可以预言，随着预测控制在理论和应用两方面的不断发展和完善，它必将在工业生产过程中发挥出越来越大的作用，展现出广阔的应用的前景。而且，目前已有公司，如 Emerson Process Management 公司，就开发有专门软件，如该公司的 DeltaV PredictPro 多变量、模型预估控制（MPC）软件，能提供实用的模型预估控制。

4.10.4　学习控制

1. 学习控制（Learning Control）概念

学习控制，面对更大的不确定性。是靠自身的学习功能来认识控制对象和外界环境的特性，并相应地

改变自身特性以改善控制性能的系统。它具有一定的识别、判断、记忆和自行调整的能力。能在其运行过程中逐步获得受控过程及环境的非预知信息，积累控制经验，并在一定的评价标准下进行估值、分类、决策和不断改善系统品质。

学习控制的学习方式分为受监视学习和自主学习两类。

受监视学习　这种学习方式除一般的输入信号外，还需要从外界的监视者或监视装置获得训练信息。所谓训练信息是用来对系统提出要求或者对系统性能作出评价的信息。如果发现不符合监视者或监视装置提出的要求，或受到不好的评价，系统就能自行修正参数、结构或控制作用。不断重复这种过程直至达到监视者的要求为止。当对系统提出新的要求时，系统就会重新学习。

自主学习　简称自学习。这是一种不需要外界监视者的学习方式。只要规定某种判据（准则），系统本身就能通过统计估计、自我检测、自我评价和自我校正等方式不断自行调整，直至达到准则要求为止。这种学习方式实质上是一个不断进行随机尝试和不断总结经验的过程。因为没有足够的先验信息，这种学习过程往往需要较长的时间。

在实际应用中，为了达到更好的效果常将两种学习方式结合起来。学习控制系统按照所采用的数学方法而有不同的形式，其中最主要的有采用模式分类器的训练系统和增量学习系统。在学习控制系统的理论研究中，贝叶斯估计、随机逼近方法和随机自动机理论，都是常用的理论工具。

2. 学习控制发展

学习控制的设想与研究始于 20 世纪 50 年代，学习机就是运用学习控制的一例，是一种模拟人的记忆与条件反射的自动装置。下棋机是学习机早期研究阶段的成功例子。

60 年代发展了自适应和自学习等方法。另一类基于模式识别的学习控制方法也用于学习控制系统。研究基于模式识别的学习控制的第三种方法是利用 Bayes 学习估计方法。

80 年代提出了反复学习控制及重复学习控制，并获得发展。

学习控制有 4 个主要功能，搜索、识别、记忆、推理。在学习系统研制初期，对搜索和识别方面研究较多，而对记忆和推理的研究还是薄弱环节。为此，傅京孙提出了需要进一步深入的课题：

（1）在非稳定环境中的学习

大多数学习算法仅在稳定的环境中有效，若把一个非稳定环境近似为若干个稳定的环境，则可应用模式识别等技术加以解决。

（2）提高学习效率

多数算法都需要较长时间，不适于快速响应系统的控制，可增加有利的先验知识加以改进。

（3）结束规则（stopping rule）

若系统已达到指定的要求，则需要有适当的结束规则，以缩短学习时间。

（4）学习系统的多级结构

对不同复杂程度的环境信息分别用不同的学习算法处理，且处于不同层次，高一级中的学习品质取决于低一级中一个或几个学习机构所获得的信息。

（5）把模糊数学用于学习系统

（6）直觉推理的应用

很多（包括复杂的）控制问题，有时只需要用直觉推理方法就可解决。

（7）文法推理

近年来，控制理论正向广度和深度发展，把人工智能技术应用于自动控制取得了可喜的成果。Saridis 提出了很多有关学习控制的新的思想方法。Astrom 等在以“专家控制”为题的开拓性论文中指出，用专家系统的方法实现工程控制中存在的很多启发式逻辑推理，可使常规控制系统得到简化，并获得新的功能。等等。

3. 学习控制结构

学习控制，最有效的途径仍是仿人和吸收人工智能的研究成果。近年来，仿人智能控制器的研究已初

见成效。智能控制算法的基本思想是仿人的学习、在线特征辨识、特征记忆、直觉推理和多模态控制策略等，而在结构上是分层的。

一个通用的仿人智能控制器（SHIC）应具有在线特征辨识的分层递阶结构，如图 4-111 所示。图中，主控制器 MC 和协调器 K 构成运行控制级；自校正器 ST 构成控制参数自校正器；自学习器 SL 构成控制规则组织级。MC、ST 和 SL 分别具有各自的在线特征辨识器 CI、规则库 RB 和推理机 IE，SL 还有作为学习评价标准的性能指标库 PB。3 个层级公用一个公共数据库 CDB，以进行密切联系和快速通信。各层级的信息处理和决策过程分别由 3 个三元序列 {A，CM，F}、{B，TM，H} 和 {C，LM，L} 描述。

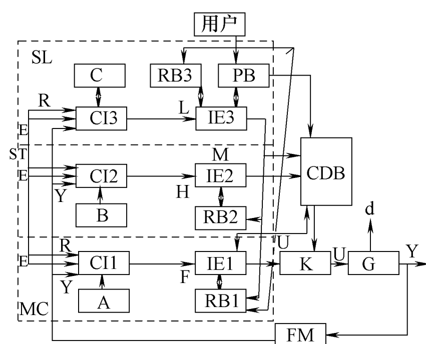


图 4-111　一个多级学习控制系统

来自指令 R、系统输出 γ 和偏差 E 等在线信息，分别送到 MC 和 ST 的 CI1 和 CI2，与相应的特征模型 A（系统动态运行特征集）及 B（系统动态特性变化特征集）进行比较和辨识，并通过 IE1 和 IE2 内的产生式规则集 F 和 H 映射到控制模式集 CM 和参数校正集 TM 上，产生控制输出 U'和校正参数 M'。U'经协调器 K 形成受控对象 G 的输入向量 U，而 M′则输入到 CDB，以取代原控制参数 M。

对于执行控制级的 MC 和参数校正级的 ST，{A，CM，F} 和 {B，TM，H} 均为由设计者赋给的或由 SL 形成的先验知识，分别存放在规则库 RB1、RB2 和 CDB 中。SL 中的 RB3 是控制器的总数据库，用于存放控制专家经验集 {C，LM，L}，它包含 {A，CM，F} 和 {B，TM，H}，选择、修改和生成规则以及学习效果的评判规则。其中，存放的性能指标包括总指标集 PA 和子指标集 PB。PA 由用户给定，PB 则为 PA 的分解子集，由 CI3 的特征辨识结果选择与组合，作为不同阶段和不同类型对象学习的依据。

学习过程分为起动学习和运行学习两种。起动学习过程是控制器起动后初始运行的学习，它反复依据当前特征状态 C，前段运行效果的特征记忆 D 以及相应问题求解的子指标集 PB 之间的关系，确定 MC 的 {A，CM，F} 和 ST 的 {B，TM，H}，即：

IF < C，D，PB >

THEN {A，CM，F} AND {B，TM，H}

运行学习过程是指控制运行中对象类型变化时的自学习过程。首先，SL 从反映对象类型变化的特征集 C′确定出新的子指标集 PB′，然后依据特征记忆 D′来增删或修改 {A，CM，F} 和 {B，TM，H}，即：

IF C′THEN PB′

IF < C′，D′，PB′ >

THEN {A′，CM′，F′} AND {B′，TM′，H′}

学习过程结束后，ST 就停止工作，处于监视状态。对于受控对象类型不变时参数和环境的不确定性变化，由 MC 和 ST 来实现快速自校正。

仿人智能控制器实时运行时，实现高品质、快速自适应和自学习控制的关键在于在线信息处理和决策的速度。为此，需要从硬件和软件两方面来解决。硬件方面除采用高速微处理芯片外，可设计并行运算的多 CPU 控制技术来支持分层递阶信息处理和决策机制。软件方面则要充分发挥特征辨识、特征记忆和直觉推理等作用，减少规则数，缩小搜索空间，以减少信息处理量。

按照上述智能控制器的设计思想，学习控制系统的设计应遵循下列基本原则：

（1）控制系统应具有分层信息处理和决策能力。

（2）控制器应具有在线特征辨识和特征记忆的功能。当被控对象的数学模型不完全清楚，且处于快速瞬变过程时，现有的系统辨识算法难以满足实时控制的要求。而在仿人智能控制中，往往只需要有限的反映受控对象特性的动态特征量，就能满足控制要求。另外，通过特征记忆，可以积累有用的信息，使控制决策更有预见性。

（3）控制器应具有多模态控制。在研究人的手动控制行为时可以发现，人的控制策略是多模态的和开闭环结合的控制方式。仿人控制具有类似的控制方式。

（4）应用直觉推理逻辑，使控制器的决策更灵活和迅速，以提高自学习效率。

4. 学习控制应用

学习控制要用到仿人智能，这将在下一小节作简要说明。而在 PLC 控制中，学习控制不一定就那么复杂，但实实在在存在着如何通过学习，提高 PLC 的控制质量及效率的问题。有这么几方面应用：

最简单的，如：十字路口的红绿灯怎么变换，才算好？可否通过学习，选定各方向的红绿灯亮、灭的合理时间分配？可否设定一个评价标准，在 PLC 实施实际控制的同时，收集数据，不断按这个标准进行评价，并依据评价结果确定转换时间。直至还可边学习，边改进。直到最合理。

再如：示教控制。可先由人工，运行程序。PLC 把人工操作记录下来。第二次，PLC 将按记录的程序，自行操作。这样的学习程序 PLC 是完全可做到的，也是较常用的。

……

4.10.5　专家控制

本章第 9 节介绍的模糊控制算法可知，模糊控制是基于人的经验的，具有“仿人智能”的特性。故也称其为是智能控制，或说它是智能控制之一。本节讨论的专家控制，则具有“高度仿人智能”的特性，是模糊控制的进一步发展，更是智能控制，或说更是智能控制之一。

1. 专家控制概念

图 4-112 所示为专家系统简化结构。它主要由知识库与推理机组成。知识库存储大量专家知识。而推理机则根据输入或提问，运用一定的推理规则，在与知识库交互中，求得与输入对应的输出，或推出提问的答案。

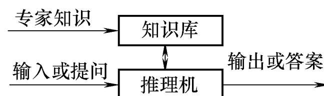


图 4-112　专家系统简化结构

专家系统知识库的数据有的为手工录入。有的可自动生成。更高级的还可自学习，不断地使用中充实、完善。

专家系统可用于种种不同的场合：如总结医生看病的经验而建立的专家系统，可进行特殊疾病诊断；再如总结设备修理的经验而建立设备诊断知识库，可协助进行设备故障诊断，等等。这些技术，在实际中得到了很好的运用，也取得了可喜的成果。

专家（Expert Control）控制则在此基础上，还要更进一步。要把专家系统和实际控制过程结合起来，用专家系统做实际控制过程的调节器。为此，专家控制就要不间断地监测被控系统状态，并运用相应的推理规则，在与它的知识库交互中，实时求得与系统状态对应的控制输出，使被控系统的工作能达到预期的效果。

可知：专家控制的知识库应是完备的，对应于每一被控系统状态，必须要有相应的控制输出答案，而这个答案必须达到人类专家的水准；专家控制的求解过程应是自动的，能根据接收到的反馈信号进行独立的、自动的决策，产生相应的控制输出；专家控制还应是实时的，要很快对反馈输入做出反应。实时性问题是专家控制应用于工业控制面临的最大困难。

为了满足工业过程的实时性要求，知识库的规模不宜过大，推理机也应尽可能简单。含有与控制有关的知识，能有效地进行推理也就够了。

在设计专家控制时应十分注意对过程在线信息的处理与利用。在信息存储方面，应对那些对作出控制决策有意义的特征信息进行记忆，对于过时的信息则应加以遗忘；在信息处理方面，应把数值计算与符号运算结合起来；在信息利用方面，应对各种反映过程特性的特征信息加以抽取和利用，不要仅限于误差和误差的一阶导数。灵活地处理与利用在线信息将提高系统的信息处理能力和决策水平。

控制策略的灵活性是对专家控制的又一要求。工业对象本身的时变性与不确定性以及现场干扰的随

机性，要求专家控制采用不同控制策略，并能通过在线获取的信息灵活地修改控制策略或控制参数，以保证获得优良的控制品质。此外，专家控制还应设计异常情况处理的适应性策略，以增强系统的应变能力。

运用专家控制，可使控制系统的设计不完全依赖于被控对象的数学模型，而主要利用人的有关知识，也可使被控对象按一定要求达到预定的控制目的。所以，在复杂的工业控制环境中，有许多未知因素和不确定因素，而利用专家知识可实现对其控制时，使用专家控制是很合适的。

2. 专家控制类型

专家控制的主要形式有二，专家控制系统和专家式控制器（Expert Controller）。前者系统结构复杂，研制代价高，因而目前应用较少。后者结构简单，研制代价低，性能又能满足工业过程控制的一般要求，因而获得日益广泛的应用。

3. 专家控制结构

专家控制系统因为应用的场合和控制要求的不同，其结构也可能不一样。然而，几乎所有的专家控制系统（控制器）都包含知识库、推理机、控制规则集和控制算法等。

图 4-113 给出了一种专家控制器的框图。

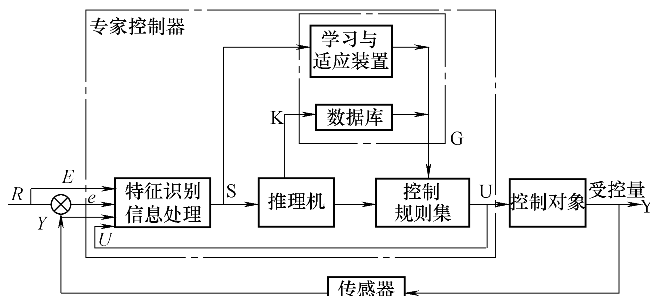


图 4-113　专家控制器

从图知，知识库由经验数据库和学习与适应装置组成。经验数据库存储经验和事实；学习与适应装置在线获取信息，补充或修改知识库内容，改进系统性能，提高问题求解能力。

控制规则集（CRS）是对被控过程的各种控制模式和经验的归纳和总结。由于规则条数不多，搜索空间很小，推理机构（IE）就十分简单，采用向前推理方法逐次判别各种规则的条件，满足则执行，否则继续搜索。

特征识别与信息处理（FR&IP）部分的作用是实现对信息的提取与加工，为控制决策和学习适应提供依据。它主要包括抽取动态过程的特征信息，识别系统的特征状态，并对特征信息作必要的加工。

专家控制器的输入集为：$E=(R,e,Y,U)$，$e=R-Y$

式中，R 为参考控制输入，e 为误差信号，Y 为受控输出，U 为控制器的输出集。

专家控制器的模型可用式 $U=f(E,K,I)$ 表示，智能算子 f 为几个算子的复合运算：$f=g\cdot h\cdot p$，其中：

$$g:E\rightarrow S;h:S\times K\rightarrow I;p:I\rightarrow U$$

g、h、p 均为智能算子，其形式为

IF A THEN B

其中，A 为前提或条件，B 为结论。A 与 B 之间的关系可以包括解析表达式、Fuzzy 关系、因果关系和经验规则等多种形式。B 还可以是一个子规则集。

上述系统当然是比较复杂的。只能运行在 PC 的平台上。PLC 使用的专家控制应力求简单、有效。其结构要简化一些。随着 PLC 性能进一步提高，较为复杂、以至于知识库可自动扩充、能自学习的专家系统会越来越多的出现的。

4. 专家控制应用

专家控制在我国已取得很大成功。正从教授、专家手中走出来，实现专家控制的工程化、实用化、转化为社会生产力。

例 1，广西大学自动化研究所已经开发出实时智能控制软件包 RICP、智能集成开放控制器 IIOC 等产品。软件包 RICP 有以下 3 个软件组成：

模糊控制系统开发工具 FCDS；

专家控制系统开发工具 ECSS；

神经网络开发工具 NNDS。

RICP 提供模糊控制、专家控制、神经网络的开发工具，可帮助用户进行模糊控制、专家控制、神经网络的设计、分析和调试，用户无须掌握深奥的智能控制理论和知识，就可以在短时间内构建智能控制系统。

RICP 采用 COM、ODBC、ActiveX、OLE、OPC，将 RICP 软件包与国内外各种工业监控组态软件（“力控”、“组态王”、“MCGS”、“世纪星”、“天工”等）无缝链接，提供软件 PLC（见第 10 章）、DCS、工控机、现场总线、以太网控制系统的接口。

RICP 人机界面友好，软硬件配套集成，图形编程组态，用户无须掌握复杂的编程知识、技术，在常规控制设备中可实现智能控制。

RICP 适用于难以建模、对象和任务复杂系统的控制，可解决工业生产过程控制的非线性，大滞后，不确定等难控问题。更多的信息可向该研究所查询。

例 2，恒压供水专家控制系统，用于城市高楼供水。它是非线性、不确定的控制过程，很难建立精确的数学模型。因此，有人将专家控制经验与传统的 PID 控制相结合，形成一套行之有效的控制算法，它的规则描述如下：

（1） IF $|e(K)| > M1 \wedge e(K) > 0$, THEN $u(K) = U1$

（2） IF $M3 < e(K) \leqslant M2 \wedge e(K) > 0$, THEN $u(K) = u(K-1) + K1e(K) - K2e(K-1)$

（3） IF $|e(K)| \leqslant M3$, THEN $u(K) = u(K-1)$

（4） IF $|e(K)| > M1 \wedge e(K) < 0$, THEN $u(K) = 0$, 关变频器

（5） IF $M3 < |e(K)| \leqslant M1 \wedge e(K) < 0$, THEN $u(K) = U2$

（6） IF $M2 < |e(K)| \leqslant M1 \wedge e(K) > 0$, THEN $u(K) = U3$

其中 $e(K) = K$ 时刻压力设定值减去 K 时刻压力实测值。U1，K1，Mi（$i = 1, 2, 3$）为现场设定值，由现场调试经验而定。$M1 > M2 > M3$。控制过程首先验证条件，条件满足则激活相应的控制策略，各控制策略之间存在一定的相互关系。

采用上述控制系统及控制规则，实现了对北京商检局及总后军需研究所的地下温泉供水系统进行变频调速恒压供水自动控制的改造，压力值的控制精度达 $\pm 0.2 kg/m^2$ 满足用户要求，节能效果显著。

例 3，小直流电动机专家控制。

系统构成主要是：直流电动机，加载给直流电动机的可调压直流电源，测量电动机转速用的光电码盘及其脉冲检测装置。具体使用装置是沈阳沈飞电子科技发展有限公司生产的 TD3 电源板、TS1、TS4 实验板、PLC 及模拟量模块元件板。

直流电动机转速由直流电源电压控制。这个电压高，电动机转速也高。两者大体为正比关系。而直流电源的电压则由 PLC 模出单元控制。本系统用的 PLC 为 CPM2A，模出单元为 CPM1A_MA002。

电动机转动将带动光电码盘转动。脉冲检测装置将检测出脉冲信号。显然，脉冲信号的频率与电动机转速是正比关系的。系统用 CPM2A 的输入点 000.00 接收这高速脉冲信号。而且，每间隔一秒计一次所采集到的脉冲数。这脉冲数也就是这个脉冲信号的频率。也就是这个频率的高低反映了电动机转速高低。

用 PLC 对电动机转速进行专家控制的目的就是，不管出现怎样的扰动，电动机转速将随给定值变化。

它用的算法是，先计算计算偏差及偏差变化率，然后，根据偏差及偏差变化率的大小，按一定规则改变控制输出，达到控制所要求的效果。整个控制效果如何？将取决专家的经验，即选定好控制周期、比较参数及变化参数。

图 4-114 ~ 图 4-117 所示的为该系统的实际程序。

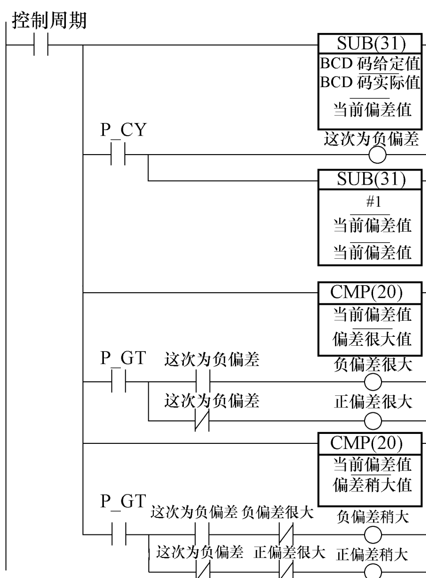


图 4-114　偏差计算

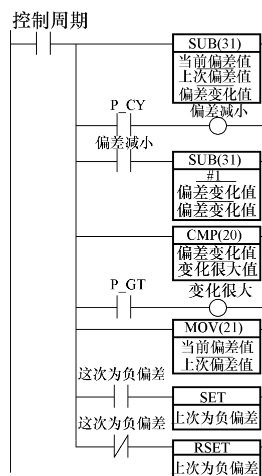


图 4-115　偏差变化率计算

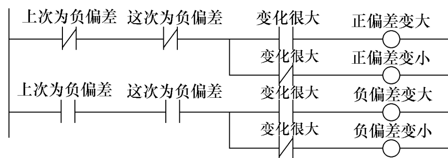


图 4-116　偏差变化分类

图 4-114 所示为偏差计算及确定偏差方向（正或负）及大小。从图知它主要用减及比较指令实现。

图 4-115 所示为偏差变化率计算。并指明变化的大小及变化方向。

图 4-116 所示的为偏差变化分类。可依据此分类确定相应的控制输出。

图 4-117 所示的依据偏差及偏差变化分类，确定控制输出应作的相应变化。输出用的是脉宽控制。而且，每一控制周期执行一次这个程序。这里的控制周期的长短，“多点”、“少点”，以及上述其它图中的“偏差很大”、“变化很大值”等的数值，可按照专家或使用中积累的经验确定。图中只画出负偏差的程序。正偏差时，情况类似。

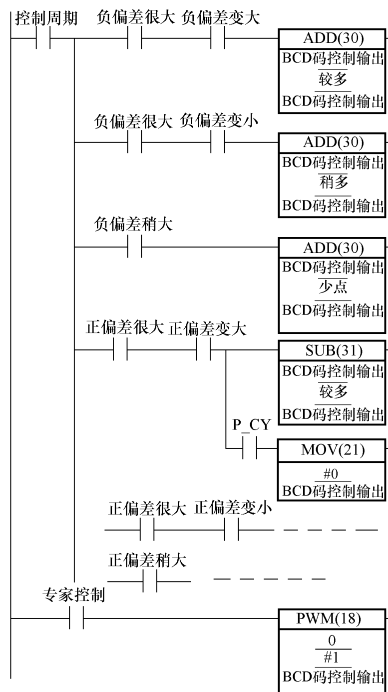


图 4-117　控制输出

结语

过程控制要用到模拟量。模拟量控制，关键要能检测到模拟量。为此，要有传感器及模拟量输入模块，用其采集模拟量，并把它转换为数字量。而又怎么把该模块转换后的数字量

读入 PLC 内存？不同 PLC 有不同的方法。OMRON 读入的为实际数，但也要分清是二进制数，还是 BCD 码。模块还需做好初始化设定等。这是其一。

其次是实现模拟量控制有效的方法是闭环控制。有时也可辅以开环。而闭环控制最有效的算法是 PID。各家 PLC 以至于同一家 PLC 不但机型不同，所用的指令也不完全一样。所以，对此要注意区别。

再，过程控制的程序实施有个参数设定问题。而这些参数合理选定与 PLC 控制系统的动力学特性有关。所以，过程控制的程序调试，只能在现场完成与完善。

本章介绍了一些高级算法可作为可供提高控制质量，及实现过程控制智能化参考。较有效的闭环控制可使用 PID 模块、温度控制模块等。如果控制回路较多，还可使用回路控制模块，或过程 CPU 及基于 PLC 的 DCS 系统。

请想想

1. 模拟量、模拟量控制含义？它与过程控制的关系？
2. 调节量（被控制量）、控制量（控制输出）的含义及其相互关系？
3. 开环与闭环控制的异同？
4. PID 算法的要点？
5. OMRON PID 指令含义与作用？
6. PID 指令用于串级控制的要点？
7. 模糊控制要点及应用？
8. 模拟量高级控制的概念？

请试试

1. 编写简单反馈控制程序。
2. 用基本运算指令编写 PID 控制程序。
3. 设计模糊控制程序。
4. 其它实际控制程序。

第 5 章　PLC 运动控制编程

在各种生产和工作机械中，运动系统是不可或缺的，因此运动控制是很常见的。本章将具体讨论运动控制的有关问题。

5.1　概述

关键词：NC、电压到频率的转换技术（V/F）、脉冲量输入（PI）、开关量输入、模拟量输入（AI）、脉冲量输出（PO）、开关量输出（DO）、模拟量输出（AO）、位置控制、运动控制、轨迹控制

运动控制大多使用脉冲量。脉冲量是取值总是不断地在 0（低电平）和 1（高电平）之间交替变化着的逻辑量。每秒钟脉冲量交替变化的次数称为频率。PLC 处理脉冲量的频率低的为几百赫、几千赫，高的为几十、几百千赫。

20 世纪 50 年代诞生的数字控制技术，简称数控（NC）技术，就是基于脉冲量的应用而不断发展与完善的。它先是应用于金属切削机床的运动控制，当今已应用到其它设备、机械手，以至于整个生产线的多方面控制，已经成为当今自动化技术的一个重要支柱。

作为后来者，已成为当今自动化又一支柱的 PLC，也可以处理脉冲量。目前，它不仅有输入、输出脉冲量的接口或模块，而且还有很多处理脉冲量的设定与指令。有的虽然为微型机、小型机，但是也有很强大的脉冲量处理功能，可以经济、有效地通过脉冲量的处理进行种种运动控制。

运动控制要用到脉冲量，但脉冲量还可以用于其它物理量的控制。其它物理量转换成电量后，利用电压到频率（V/F）的转换技术，也可以再转换成脉冲量。因而，脉冲量也可用以检测其它物理量，如温度、湿度、流量。此外，脉冲量的输出还可以实施脉宽调制（PWM），因而脉冲量也可用以连续地改变控制输出。有了 V/F 转换、PWM 这 2 项技术，脉冲量用以实现模拟量的控制也成为可能。而且用脉冲量实现控制的还有如下优点：

（1）系统的工作精度高，并且这个精度可以得到控制。因为它的精度是以脉冲计算。减少脉冲当量，即减少每脉冲对应的物理量的实际值，就可以提高控制精度。随着技术进步，脉冲当量可以做得非常小。

（2）资源比较节省，它是串行处理数值，而不是并行处理数值。用一个输入（单相时）或输出点（一个位）就可以处理原来用一个通道（16 位）或字节（8 位）才能处理的问题。这里的关键是，当今 PLC 的工作速度很高，可赢得时间。有了时间的富裕，就可以换取这个空间的节省。

（3）它的信号传送的电平高，信号失真对其影响也较小，所以抗干扰能力很强。

用脉冲量实现控制有以上 3 个优点，所以近年来它不仅用于运动控制，而且在对其它

的物理量的控制上，也用得越来越多。

5.1.1 运动控制的目的

运动控制的目的有以下6个：

1. 位置控制

位置控制是指控制对象的位置移动，也称为定位控制。例如用立体仓库的操作机取货、送货，首先就要定位，即要移动到指定的位置，其次才能进行相关操作。

位置控制可以用闭环控制，也可以用开环控制。

闭环控制总是要得知反映位置的脉冲量变化，并依此确定进一步的控制，具体是脉冲量输入（PI）和开关量输出（DO）。脉冲量输入是读入的脉冲，读入后与设定值（控制要求）比较，再根据比较结果确定相应的开关量输出（DO）或模拟量输出（AO），进而实现位置控制。

开环控制可以不管脉冲量怎么变化，总是按照原定的目标进行控制。具体是开关量输入（DI）和脉冲量输出（PO），也可以按照设定的程序（预定要求）输出脉冲。开环控制要用到PLC的脉冲输出功能（或使用脉冲输出点，或使用位置控制单元），要按照要求输出脉冲。

2. 轨迹控制

它用于多个坐标的运动控制，不仅要控制目标位置，控制运动规律，还要控制坐标间的位置协调。其目的是使对象能按照要求的轨迹运动。

基于计算机技术的NC是实现这个要求的最好方法，用它可以实现复杂型面的单件、小批量生产的自动化，已经发展到非常完善的境地。NC系统功能很强，例如NC系统可以实现5个坐标的协调控制。还有很多其它功能。但是它很复杂，价格昂贵。

PLC也可实现NC的某些功能，如目前也可以实现多个坐标协调控制，数据长度可能短些，但是比NC要简单得多，价格比NC也低得多。

特别在近期，不少PLC厂商推出各种运动控制单元，不仅可进行2轴或多轴的位置控制，可以形成各种曲线轨迹，还可以用数控的G语言编程（要另用有关编程器或编程软件），而且不需要梯形图，可存储的程序也多，操作更加方便，为在PLC中使用NC技术开辟了新的前景。

3. 速度控制

它不仅控制目标位置，而且要控制运动规律。这些规律是指有确定的起动速度、加速度、运行速度、减速度、结束速度等。

位置控制再加上运动控制，可以实现更为准确的定位。

速度控制可以是开环，也可以是闭环。

4. 转矩控制

转矩控制的目的是使控制电动机的输出转矩的大小相对衡定。如设定某个转矩，如果电动机轴负载低于这个设定转矩时电动机快转；外部负载等于这个设定转矩时电动机慢转，大于这个设定转矩时电动机不转或反转。

转矩控制主要应用在对材质的受力有严格要求的卷取和开卷的装置中，例如绕线装置或拉光纤设备。而这个设定转矩还要根据卷取半径的变化随时更改，以确保材质的受

力不会随着卷取半径的变化而改变。

5. 同步控制

同步控制要求运动对象运动保持同步。如果用开环控制，设法保持各个坐标按比例输出脉冲就可以了。如果用闭环控制，实际是随动控制或为称跟踪控制。应使其它坐标的运动能跟上目标坐标的运动，以实现坐标间运动同步。如使火炮炮身运动能跟随雷达天线的运动所进行的控制，就是典型的随动控制。闭环控制的办法是时时检测目标运动的位移量，并转换为脉冲量，以作为其它坐标的输出脉冲。

6. 模拟量控制

以上 5 个目的，都是针对运动量控制的，这也是早期脉冲量控制的目的。如今，随着脉冲技术的发展，脉冲生成器、执行器的不断开发与完善，用部分脉冲量（输入或输出有一个用脉冲量），或全部用脉冲量（输入和输出全用脉冲量），对其它模拟量进行控制，包括用基本反馈控制、PID 控制、模糊控制及其它智能控制也都可以实现。有关的目的，如保持被控量定值（定值控制）、使被控量跟随别的量变化（随动控制）或按一定程序变化（程序控制）等，也都可以实现。

5.1.2　运动控制的类型

PLC 运动控制类型，按是否需要反馈信号区分，有开环控制与闭环控制，此外还有同步控制。

1. 开环控制

开环控制的控制量采用串行脉冲信号，执行器使用步进电动机或伺服电动机。执行器按接收到脉冲数运动。接收到多少脉冲，产生多少位移，一般是不会丢失脉冲的，所以不要求输出结果的反馈，也可保证运动精度。

开环控制无需反馈，只需不多输入量，总是按预先设计好的程序，一步步输出控制脉冲。一旦起动控制命令，运动对象将按要求的速度、加速度或轨迹运动，直到预定的控制任务完成。所以，开环控制实际是程序控制。

开环控制比较简单，既可实现复杂运动的控制，又能保证足够的控制精度。所以，在当今运动控制中，它是用得最多的控制。只是由于开环控制要使用脉冲量，所以要求 PLC 的工作速度要很高。

2. 闭环控制

闭环控制不仅需要控制量，还需要反馈量。第 4 章讨论的过程控制多数是闭环控制，但在运动控制中也有所应用。

PLC 输入、输出的数据类型有：脉冲量输入（PI）、开关量输入（DI）、模拟量输入（AI）、脉冲量输出（PO）、开关量输出（DO）及模拟量输出（AO）6 种。与此对应，与运动量及脉冲量相关的闭环控制大体有这么几种可能的组合：

（1）脉冲量输入（PI）、开关量输出（DO）。这种控制反馈输入是脉冲量，控制输出用开关量。PLC 计入接收到的脉冲，不断地与设定值比较，并根据比较结果产生开关量输出进行控制。

（2）脉冲量输入（PI）、模拟量输出（AO）。这种控制反馈输入是脉冲量，控制输出用模拟量。PLC 计入接收到的脉冲，并转换为脉冲频率。用它代表反馈模拟量输入的大

小。这个反馈量与给定量一起，按照一定算法（如 PID）处理，可得到控制量，然后再通过模拟量输出进行控制。

（3）脉冲量输入（PI）、脉冲量输出（PO）。这种反馈输入、控制输出都是脉冲量。PLC 计入接收到的脉冲，并转换为脉冲频率。用它代表反馈模拟量输入的大小。这个反馈量与给定量一起，按照一定算法（如 PID）处理，可得到控制量。然后，用 PWM 输出进行控制。

（4）模拟量输入（AI）、脉冲量输出（PO）。这种反馈输入是模拟量，控制输出用脉冲量。PLC 读入反馈模拟量。它与给定量一起，按照一定算法（如 PID）处理，可得到控制量，然后用 PWM 输出进行控制。

（5）开关量输入（DI）、脉冲量输出（PO）。多位开关量输入点连接绝对值旋转编码器，可用以读入运动部件位置现值。这个现值与要求到达的位置值比较或处理，产生控制脉冲输出进行控制。

（6）开关量输入（DI）、模拟量输出（AO）。多位开关量输入点连接绝对值旋转编码器，可用以读入运动部件位置现值。这个现值与要求到达的位置值比较或处理，产生模拟量输出进行控制。

（7）开关量输入（DI）、开关量输出（DO）。多位开关量输入点连接绝对旋转编码器，可用以读入运动部件位置现值。这个现值与要求到达的位置值比较或处理，产生开关量输出进行控制。

本章将针对这些组合，对闭环控制作进一步讨论，算是对第 4 章讨论的补充。

3. 同步控制

在运动控制中，同步控制也很常见。实现同步控制可以用开环，也可以用闭环，本章也将简要讨论这些控制。

5.1.3 运动控制的特点

运动控制多是用脉冲量，与控制开关量、模拟量不同，有自己的特点，这些是：

1. 控制功能设定多

PLC 用于开关量控制，什么设定都可以不做。PLC 用于模拟量控制，要有一些设定，如输入、输出的信号量程等，但是不多。而使用 PLC 的脉冲量控制功能，要设定的项目则很多。

如高速计数（读入高速脉冲信号），多数小型 PLC 可以使用具有高速计数功能的输入点接收脉冲。但是在使用之前，首先要设定是否使用这个功能？接着还要设定用什么模式（方式）进行高速计数？计数怎么复位？再如中、大型机，虽然无这样的输入点，但是要用高速计数特殊单元。而且使用它，也要作相应设定，如机号是否启用（如多路时，有的路可以不用）、使用的计数方式（模式）、复位方式等等。

再如脉冲输出，多数小型 PLC 可以使用具有脉冲输出功能的输出点输出脉冲。但是在使用之前，有的也要设定。如设定输出什么脉冲？是脉宽调制的，还是标准的？再如，中、大机，无这样的输出点。但是可以使用位置或运动控制特殊单元。使用它，也要作相应设定。

2. 控制参数选定多

除了设定，在脉冲控制之前，要选用的控制参数还很多。

如高速计数，定时器的现值常要与目标值进行比较，而且这个目标值多不是一个，而是很多个。这样才能依计数达到不同的目标值，实施不同的控制。这些目标值各是多少？就是参数选定。

再如脉冲输出，也有很多控制参数，须根据工艺要求合理选定。

很多的参数选定，对编程增加不了多大的工作量，但是将给程序的现场调试增加很多工作量。

3. 开环控制用的多

脉冲量控制，特别是用脉冲量进行运动控制，多数是开环控制。开环控制只是发命令，不需要信号反馈，比较简单。模拟量开环控制也简单，但不精确。而用脉冲量开环控制则是既简单又精确，只要用机械的方法确保一个脉冲，前进一个步距，且步距很小，实现可靠、精确的控制是有保证的。事实上这种机械实现的方法很多，并且已经有很多完善的系统。

4. 时延较长

用脉冲量进行闭环控制时，特别是用脉冲做反馈信号，时延是较长的。因为脉冲读入有个过程。如检测脉冲频率，仅读一个脉冲是无法测算其频率的，须有一个时间间隔，在这个时间间隔中，观察读入多少脉冲，再用读入的脉冲数，除以时间间隔，才能得知它的频率。显然，经历这些过程是需要时间的，这个时间也就造成了系统控制较大的时延。

5. 中断使用多

在顺序控制、过程控制的编程中，使用中断程序是不多的。这是因为 PLC 的正常扫描工作速度就足以满足要求了。但是利用脉冲量工作的运动控制，由于脉冲工作频率高，为了确保 PLC 的输出，能对脉冲输入有很快的响应速度及提高系统工作精度，不得不使用 PLC 的很多中断技术。

6. 编程方法多

运动控制编程方法多。小型机可用系统提供的指令编程，如脉冲输出、进行高速计数，都有相应指令可供选用。

中、大型 PLC，根本无此类指令，它的脉冲读入、输出，全靠特殊单元自身控制。PLC 要编的程序，只是对相应的控制位，进行置位、复位操作及作些参数传送就可以了。

有些中大型 PLC 运动控制单元，自身还有编程软件。可以用自身的语言，如使用数控用的 G 语言进行编程。这样的系统，PLC 要编的程序就更少了，最多只是编写与特殊单元的数据交换及交换前后的数据处理的有关程序。

所以，本章讨论对脉冲量控制程序设计，很多的视线是集中在功能设定、参数选定上。也顺便提醒读者，在 PLC 脉冲量控制编程时，应首先把注意力集中在功能设定、控制参数的选定上。为此，详细阅读有关说明书，全面了解 PLC 及有关模块在实施脉冲量控制方面的特性，具体弄清有关工艺要求与控制参数选定的关系，就非常重要了。

不过，如果用小型机利用脉冲量进行多坐标协调控制，程序还是比较复杂的。为此，本章对这些控制程序作了较详细的介绍，以帮助读者能熟悉这方面的编程。

此外，用 PLC 实现模拟量控制的其它特点，如有误差、断续（离散），在脉冲量控制中也是存在的。

5.2　脉冲量控制硬件基础

关键词：旋转编码器、脉冲信号编码器、二进制码盘、格雷码、直线编码器、电压频率转换器、频率电压转换器、步进电动机、伺服电动机。

5.2.1　脉冲信号生成

运动控制要用到脉冲信号。脉冲信号编码器是用以生成脉冲信号的装置。它们把模拟信号，如位移、速度、温度，转换成脉冲信号。具体有旋转编码器、感应同步器、温度脉冲转换器、湿度脉冲转换器，等等。

1. 编码器

它是测量角位移的数字编码器。具有分辨能力强、测量精度高和工作可靠等优点，是测量轴转角位置的一种最常用的位移传感器。编码器可分为绝对值编码器和增量式编码器两种，前者能给出角向位置绝对值；后者只能给出角向位置相对值。

（1）绝对值编码器。绝对值编码器可分为接触式和非接触式两种。接触式绝对编码器由编码盘、电刷和电路组成。图 5-1 所示为接触绝对值编码器的编码盘示意图。

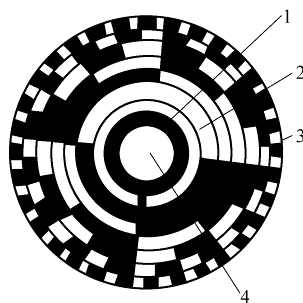


图 5-1　二进制的 6 位接触绝对值编码器的编码盘
1—激励轨道　2—最高位轨道
3—最低位轨道　4—此位置时输出为 1101010

它是一个 6 位二进制编码器。码盘有 6 条码道，按二进制编码。每条码道上有许多扇形导电区（黑区）和不导电区（白区），全部导电区连在一起接到一个公共电源上。6 个电刷沿一个固定的径向安装，分别与 6 条码道接触。每个电刷与一单根导线相连，输出 6 个电信号，其电平由编码盘的位置决定。当电刷与导电区接触时，输出为“1”电平；与不导电区接触时，输出为“0”电平。随着转角的不同，输出相应的编码。编码器的精度取决于编码盘本身的精度，分辨率取决于码道的数目。如 10 条码道，其分辨率为 1/1024。

二进制码优点是可直接进入计算机工作，但它在交界面上会出现错读，并且随着码盘输出值的增加，读数误差也伴随增大。例如在图 5-1 二进制码盘中，0 与 15 的交界面上，由于工艺和装配的因素可能读成 1111 或 0000 以外，任何数字都可出现，即发生非单值性，这就产生读数出错。

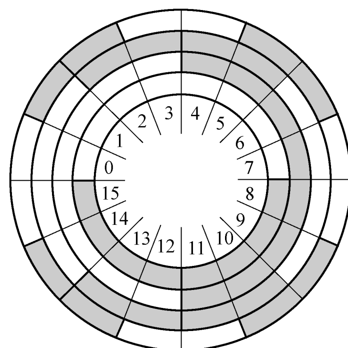


图 5-2　循环码编码盘

克服这个缺点可采用循环二进制码（图 5-2 所示为循环码编码盘），又称格雷码。其计数变化时仅一个位发生变化，所以输出不会出错。循环码在结构上还有

一个很大优点是，最低位区段的长度比二进制码区段长度大一倍，即在同样条件下，循环码编码盘的精度比二进制码编码盘高一倍，或者在相同精度和工艺条件下，循环码编码盘直径要小一半。

格雷码是无“权”值的码，不能直接代表实际数值，输入PLC后，须经译码，PLC才能判断大小。它与十进制数的关系为

$$D = +1\sum_{j=0}^{n}2^{j} - 1\sum_{j=0}^{m}2^{j} + 1\sum_{j=0}^{q}2^{j} - 1\sum_{j=0}^{s}2^{j} + \cdots$$

式中D为十进制数，n为具有“1”输出的最高位的位数，m为其次一位具有“1”输出的位数，q、s、…依次类推。

格雷码与二进制码也可相互转换。采用格雷码时，还需先进行转换，然后才能运算。

接触绝对式编码器的缺点是编码盘与电刷之间存在接触摩擦，使用寿命短。可用另一种非接触绝对式编码器。即光学绝对式编码器，它是依照光学和光电原理制成的器件。它由光源、编码盘、光学系统及电路4部分组成。如图5-3所示。

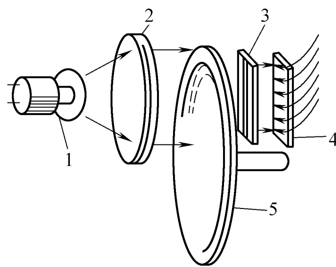


图5-3　光学编码器原理示意图
1—光源　2—透镜　3—窄缝
4—光电元件　5—编码盘

编码盘是在不透明的基底上按二进制码制成透明区图案，相当于接触编码器的导电区和不导电区。光线通过编码盘由光电元件转换成相应的电信号。

（2）增量式编码器。它比绝对式编码器使用的编码盘轨道少，这样它的导线数、滑环数、读出器、电路和显示元件保持最低，使得系统可靠性增大，成本降低。因此，现代系统多倾向于采用增量式编码器。增量式编码器主要缺点是测量相对于一个基准数，假如这个基准数有误差，整个系统受影响；另一个问题是当电源出现故障时，常常导致数据丢失，须使用辅助数据记忆技术，以防止丢失。

目前，这两种都有现成的产品，市场上即可购得，按说明书接线及使用即可。

2. 光栅式传感器

它是采用光栅叠栅条纹原理测量位移的传感器。光栅是在一块长条形的光学玻璃上密集等间距平行的刻线，刻线密度为10～100线/mm。由光栅形成的叠栅条纹具有光学放大作用和误差平均效应，因而能提高测量精度。

传感器由标尺光栅、指示光栅、光路系统和测量系统4部分组成，如图5-4所示。

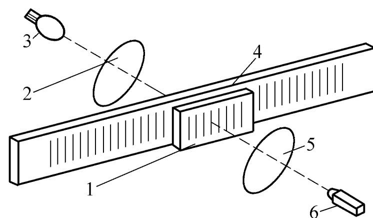


图5-4　光栅式传感器工作原理图
1—指示光栅　2、5—透镜　3—光源
4—标尺光栅　6—光电元件

标尺光栅相对于指示光栅移动时，便形成大致按正弦规律分布的明暗相间的叠栅条纹。这些条纹以光栅的相对运动速度移动，并直接照射到光电元件上，在它们的输出端得到一串电脉冲，通过放大、整形、辨向和计数系统产生数字信号输出，直接显示被测的位移量。

传感器的光路形式有两种：一种是透射式光栅，它的栅线刻在透明材料（如工业用白玻璃、光学玻璃等）上；另一种是反射式光栅，它的栅线刻在具有强反射的金属（不锈钢）或玻璃镀金属膜（铝膜）上。

这种传感器的优点是量程大和精度高。

3. OMRON 公司生产的旋转编码器

OMRON 公司是旋转编码器生产的大厂商。绝对式的、相对式的都有很多型号。型号不同，分辨率、产品结构、防护技术及接线方法也有所不同，价格也相差很大。

OMRON 公司还生产直线编码器，型号也较多。图 5-5 所示为直线编码器外观。

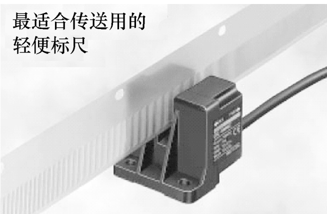


图 5-5 直线编码器

4. 电压频率变换器

V（电压）F（频率）C（变换器）与 F（频率）V（电压）C（变换器）用于电压与频率信号互换。有不少生产厂家生产此产品。国内如青岛晶体管实验厂就也研制和生产此类产品，有 QD 系列及 4602 型 1MHzVFC、8107 型 500kHzVFC、8106 型高线性 VFC、4502 型单电源 VFC 等，共有 16 个型号。

5.2.2 脉冲信号接收

对于接收脉冲信号，中、大型 PLC 用高速计数单元，小型 PLC 用高速计数输入点，其它输入点也可处理，只是可以处理的脉冲频率低些。

1. 小型 PLC 高速计数输入

小型 PLC 多有可处理脉冲量的 I/O 点。高速计数时，每组一般有 3 个输入点。如 CPM2A PLC 可使用 000 通道的 00、01、02 3 个输入点处理脉冲量输入，进行高速计数。新型的小型 PLC 的这个功能及性能又有很大提升。如 CQM1H 型 PLC 还有高速计数的内插板，插入 CPU 单元的内，可处理 4 路、频率高达 500kHz 的高速计数。再如 CJ1M 机内置有多个脉冲输入、输出点。可进行多组、多种脉冲输入、输出操作。CP1H 机是 OMRON 公司新推出的小型 PLC，也有很强的高速计数功能。以下以 CP1H 机为例对小型 PLC 高速计数输入作简要介绍：

（1）高速计数模式。可按不同特点分类：

1）以计数信号分

（a）两相输入。有 A、B 及 Z 三相，用编码器输入脉冲，可能的一种接线如图 5-6 所示。

当 A 相超前于 B 相 90°为增计数，反之为减计数。超前及滞后与旋转编码器的转动方向有关。这正反映了实际运动的情况。A、B 相信号波形如图 5-7 所示。

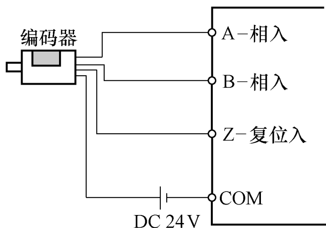


图 5-6 编码器接线

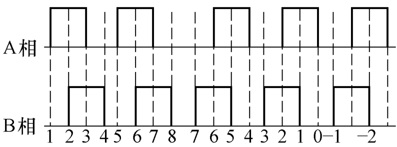


图 5-7 A、B 相信号波形

从图 5-7 可知，一个脉冲周期，其输入的脉冲数为 4。计算其转动量及最高频率时，一定要考虑到这一点。编码器每旋转一圈发一个脉冲信号。Z 相为复位信号，如需硬件复位时，要用到它。

（b）脉冲 + 方向输入。使用方向信号输入及脉冲信号输入，根据方向信号的状态（OFF/ON）将计数值相加或相减。

（c）加减脉冲输入。有两个输入点，一个输入正向脉冲，增计数，一个输入反向脉冲，减计数。

（d）单相输入。单相输入仅用一个脉冲输入端。有脉冲入，计数值即增加。

2）以计数范围分

（a）线性模式。在下限（-2147483648）与上限（2147483647）值之间计数。初始计数从 0 开始，如超过上限，则溢出；如超过下限，则下溢，并都停止计数。如果为加法模式计数，则下限为 0，上限为 4294967295（FFFFFFFF）。

（b）环形模式。在设定范围内对输入脉冲进行循环计数。如加计数，到最大值，则归 0 后再继续计数。如减计数，减到 0，则先变为最大值，再继续减计数。设定的最大值不应超过二进制值 FFFFFFFF。用环形模式计数，不存在负值。

3）以复位方式分

（a）软件复位。高速计数器复位标志 OFF→ON 时，将高速计数器当前位置复位，但要在赶上 I/O 刷新，即一个扫描周期后，才能实现复位，如图 5-8 所示。

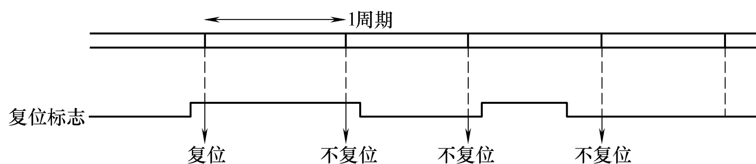


图 5-8　软件复位

（b）Z 相信号 + 软件复位。高速计数器复位标志为 ON 的状态下，Z 相信号（复位输入）OFF→ON 时，将高速计数器当前值复位。它常用于对编码器做多圈计数的场合。图5-9 所示为这种复位方式的定时图。

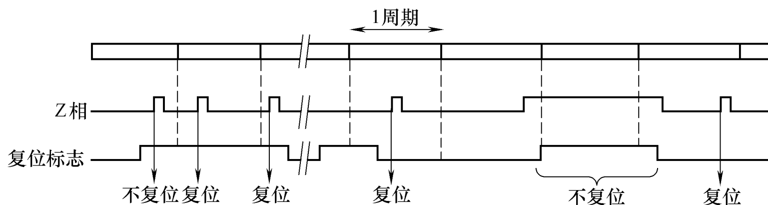


图 5-9　Z 相信号 + 软件复位定时图

此外，还可选定复位后高速计数器比较是否继续。

（2）高速计数特性。CP1H 机有 X/XA（内嵌有模拟量输入输出点）及 Y 型机。其高速计数特性见表 5-1。高速计数使用的内存区见表 5-2。

表 5-1 高速计数特性

<table>
<tr><th colspan="4">项　目</th><th colspan="4">内　容</th></tr>
<tr><td colspan="4">高速计数器点数</td><td colspan="4">4点(高速计数器0~3)</td></tr>
<tr><td colspan="4">计数器模式
(依据PLC系统设定进行选择)</td><td>相位差输入</td><td>加减法脉冲输入</td><td>脉冲+方向输入</td><td>加法脉冲</td></tr>
<tr><td colspan="4" rowspan="3">输入引脚编号</td><td>A相输入</td><td>加法脉冲输入</td><td>脉冲输入</td><td>加法脉冲输入</td></tr>
<tr><td>B相输入</td><td>减法脉冲输入</td><td>方向输入</td><td>-</td></tr>
<tr><td>Z相输入</td><td>复位输入</td><td>复位</td><td>复位输入</td></tr>
<tr><td colspan="4">输入方式</td><td>相位差4倍频(固定)</td><td>单相输入×2</td><td>单相脉冲+方向</td><td>单相脉冲</td></tr>
<tr><td rowspan="3">响应频率</td><td>X/XA型</td><td>高速计数器0~3</td><td>DC 24V输入</td><td>50kHz</td><td>100kHz</td><td>100kHz</td><td>100kHz</td></tr>
<tr><td rowspan="2">Y型</td><td>高速计数器0、1</td><td>线路驱动器输入</td><td>500kHz</td><td>1MHz</td><td>1MHz</td><td>1MHz</td></tr>
<tr><td>高速计数器2、3</td><td>DC 24V输入</td><td>50kHz</td><td>100kHz</td><td>100kHz</td><td>100kHz</td></tr>
<tr><td colspan="4">数值范围模式</td><td colspan="4">线性模式、环形模式(通过PLC系统来设定)</td></tr>
<tr><td colspan="4">计数值</td><td colspan="4">线性模式时:80000000~7FFFFFFFHeX
环形模式时:00000000~环形设定值
(在00000001~FFFFFFFFHeX的范围内,通过PLC系统设定来设定环形设定值)</td></tr>
<tr><td colspan="4" rowspan="2">高速计数器当前值保存目的地</td><td colspan="4">高速计数器0:A271CH(高位)/A270CH(低位)
高速计数器1:A273CH(高位)/A272CH(低位)
高速计数器2:A317CH(高位)/A316CH(低位)
高速计数器3:A319CH(高位)/A318CH(低位)
对于该值,可进行目标值一致比较中断或区域比较中断
注:共同处理的时间内每周期被更新
　读取最新值的情况下,使用PRV指令</td></tr>
<tr><td colspan="4">保存数据形式:十六进制8位(BIN)
线性模式时:80000000~7FFFFFFFHex
环形模式时:00000000~环形设定值</td></tr>
<tr><td colspan="2" rowspan="2">控制方式</td><td colspan="2">目标值一致比较</td><td colspan="4">登录48个目标值及中断任务号</td></tr>
<tr><td colspan="2">区域比较</td><td colspan="4">登录8个上限值、下限值、中断任务号</td></tr>
<tr><td colspan="4">计数器复位方式
(依据PLC系统设定进行选择)</td><td colspan="4">● Z相信号+软复位
复位标志为ON时,通过Z相输入的ON进行复位
● 软复位
通过复位标志为ON进行复位
注:将高速计数器复位时,可选择停止或继续比较动作</td></tr>
</table>

■ 区域分配

表 5-2　高速计数使用的内存区

内　　容		高速计数器 0	高速计数器 1	高速计数器 2	高速计数器 3
当前值保存区域	保存高位 4 位	A271CH	A273CH	A317CH	A319CH
	保存低位 4 位	A270CH	A272CH	A316CH	A318CH
区域比较一致标志	与比较条件 1 相符时为 ON	A274.00	A275.00	A320.00	A321.00
	与比较条件 2 相符时为 ON	A274.01	A275.01	A320.01	A321.01
	与比较条件 3 相符时为 ON	A274.02	A275.02	A320.02	A321.02
	与比较条件 4 相符时为 ON	A274.03	A275.03	A320.03	A321.03
	与比较条件 5 相符时为 ON	A274.04	A275.04	A320.04	A321.04
	与比较条件 6 相符时为 ON	A274.05	A275.05	A320.05	A321.05
	与比较条件 7 相符时为 ON	A274.06	A275.06	A320.06	A321.06
	与比较条件 8 相符时为 ON	A274.07	A275.07	A320.07	A321.07
比较动作中标志	与比较条件实行中为 ON	A274.08	A275.08	A320.08	A321.08
溢出/下溢标志	线性模式中,当前值溢出或下溢时为 ON	A274.09	A275.09	A320.09	A321.09
计数方向标志	0:减法计数时 1:加法计数时	A274.10	A275.10	A320.10	A321.10
复位标志	用于当前值的软复位	A531.00	A531.01	A531.02	A531.03
高速计数器选通标志	选通标志为 1(ON),禁止脉冲输入的计数	A531.08	A531.09	A531.10	A531.11

（3）高速计数器使用。首先要做好设定，确定计数模式、计数范围及复位方式；其次，要做好接线；最后，编写及调试程序。

1）设定。用 CX-Programmer 编程软件，在设定窗口的“内置输入设置”画面上，如图 5-10 所示。对高速计数器 0 ~ 3 进行设定。设定后下载给 PLC。

从 5-10 图可知，在该画面上可对是否使用高速计数器，以及使用的计数模式、输入模式及复位方式等进行设定。设定的范围见表 5-3。



图 5-10　高速计数器设定画面

2）接线。图 5-11 所示为 CP1H 机高速计数器输入点布置图。如高速计数器 0，其输入点为 08、09、03 点。其功能图上也有注明。接线就要注明接。

图 5-12 所示为使用 OMRON E6B2-CWZ6C NPN 开路输出的编码器与 PLC 输入点接线。用的 24V 直流电源。

表 5-3 高速计数器设定范围

项　　目	设 定 内 容	项　　目	设 定 内 容
高速计数器 0 ~ 3	使用	复位方式	Z 相信号 + 软件复位(比较继续)
数值范围模式	线形模式		软件复位(比较继续)
	循环模式	计数模式	相位差输入(4 倍频)
环形计数器最大值	0 ~ 4.294,967,295		脉冲 + 方向输入
复位方式	Z 相信号 + 软件复位		增减脉冲
	软件复位		递增脉冲

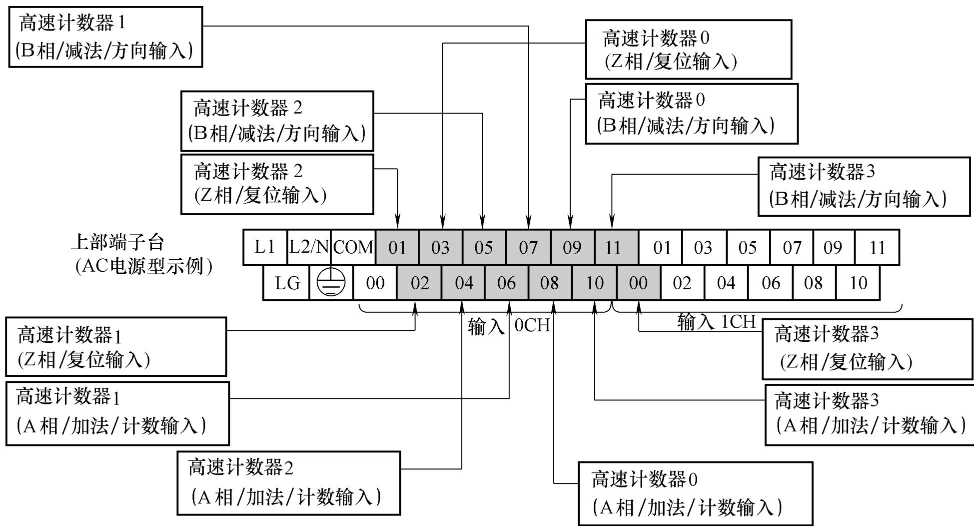


图 5-11 CP1H 机高速计数器输入点布置

3）编程。要在主程序中调用高速计数处理指令，同时还要编写高速计数中断服务程序。

（a）高速计数处理指令。有 INI（动作控制）、PRV（当前值读取）、PRV2（频率转换）及 CTBL（表比较）4 个指令。

a）INI 指令。可进行如下动作控制：开始或停止高速计数器比较，变更高速计数器当前值，变更中断计数器输入当前值。此外还可控制脉冲输出。其梯形图符号为：

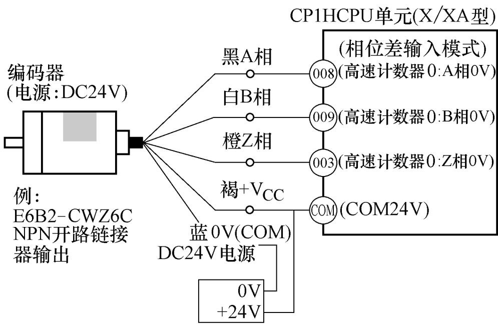
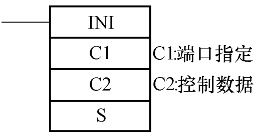
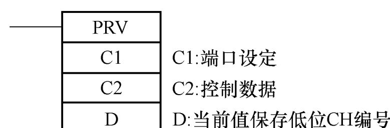


图 5-12 E6B2-CWZ6C 编码器与 PLC 输入点接线



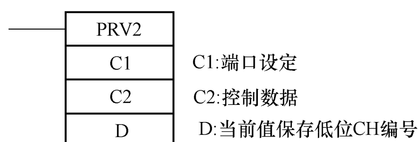
这里，C1 指定输入或输出端口，十六进制数 0000 为脉冲输出 0、16 进制数 0001 为脉冲输出 1、十六进制数 0010 为高速计数器 0、十六进制数 0011 为 1，十六进制数 1000 为 PWM 输出 0、十六进制数 1001 为 PWM 输出 1，等等（具体因 PLC 型号而变）。C2 为控制字，十六进制数 0000 为开始比较，十六进制数 0001 为停止比较，十六进制数 0002 为待变更的当前值，十六进制数 0003 为停止脉冲输出（在脉冲输出停止状态下将清除脉冲量设定）S、S+1 两个字用以存储待变更的当前值。不是变更功能设为十六进制数 0000，不使用。

b）PRV 指令。主要用以读取高速计数器及脉冲输出当前值、频率、状态（是否溢出、是否进行比较）及比较结果。梯形图符号为



这里，C1 指定输入或输出端口，同 INI 指令。D、D+1 两个字用以存储读取值。C2 为控制字，十六进制数 0000（0000Hex）为读取当前值，0001Hex 为读取状态，0002Hex 为读取比较结果，00□3Hex 为读取脉冲频率（□=0，通常方式；□=1，10ms 采样方式；□=0，100ms 采样方式；□=0，1s 采样方式）。通常方式用每个脉冲计数时间的计数进行计算，但在高频时，脉冲上升沿/下降沿失真会引起误差（参考：100kHz 时最大误差为 1%。1MHz 时的最大误差为 5%，指定时间采样方式是用测量一定时间（采样时间）内的计数脉冲、计算频率，可选择 3 个时间，即具体为 10ms、100ms 及 1s（计算该时间内的脉冲数）。

c）PRV2 指令。把读取高速计数器中脉冲频率转换成转速，或把高速计数器的当前值转换成累计旋转数，并用 8 位十六进制数存储，但只能在高速计数器 0 中使用。其梯形图符号为



这里，C1 指定转换项目，十六进制数 1 为当前值到累计旋转数转换，十六进制数 0#*0 为频率旋转速度转换，其中#指定转速单位、*指定频率计算方式。C2 指定系数。D、D+1 两个字用以存储转换结果值。

d）CTBL 指令。用以登记与起动高速计数器当前值与设定值的比较。本指令执行一次即有效。其梯形图格式为



这里，C1 指定输入或输出端口，十六进制数 0000 为高速计数器 0，十六进制数 0001 为高速计数器 1、十六进制数 0002 为高速计数器 2、十六进制数 0003 为高速计数器 3。C2 为

控制字，十六进制数 0000 为登录并起动目标值比较，十六进制数 0001 为登录并起动目标范围比较、十六进制数 0002 为登录目标值比较、十六进制数 0003 为登录目标范围比较。S 开始的若干字用以存储有关比较参数。

如果为目标值比较，这里 S 开始的若干字含义为：

S 指定多少个比较目标值，可在十六进制数 0001 ~ 0030 之间选取。

每个目标值比较固定占用 3 个字。第一个、第二个字为比较目标值，可以在十六进制数 0000 ~ FFFF 之间选取，如使用多个目标值比较，这些目标值不能相同，同时不能把计数的上下限值作为目标值。第三个字指定增或减计数有效及中断任务号。其中高字节的最高位指定增、减计数有效，0 为增计数有效，1 为减计数有效；低字节指定中断任务号，可在十六进制数 00 ~ FF 之间选取。

增减计数有效含义为，如设定增计数有效，只要高速计数器当前值增加到与目标值相等，则激发所指定的中断任务。如设定减计数有效，只要高速计数器当前值减少到与目标值相等，则激发所指定的中断任务。

这个目标值比较设定有多少字，取决于在 S 中指定。最多可扩展到 S + 142 ~ S + 144。

如果为目标范围比较，固定使用 S ~ S + 39 共 40 个字。每个目标范围比较固定占用 5 个字，所以最多可进行 8 个范围比较。这 5 个字的含义为：

第一个、第二个字为比较范围下限，可以在十六进制数 0000 ~ FFFF 之间选取。第三个、第四个字为比较范围上限，可以在十六进制数 0000 ~ FFFF 之间选取，但必须大于下限。第 5 个字指定中断任务号。只要高速计数器当前值落入比较范围（含上、下限值），则激发所指定的中断任务。中断任务号，可在十六进制数 00 ~ FF 之间选取。

如果不进行那么多比较，可把不进行比较的中断号设为十六进制数 FFFF。

（b）程序实例。

a）图 5-13 所示为读脉冲频率程序。当 0.01 为 ON 时，执行 PRV 指令，在该状态下读取输入到高速计数输入 0 中的脉冲频率，由十六进制数输出到 D200 中。

提示：这里的 C1、C2 值，输入数字前要加“#”号，但早期机型不加“#”。以下各指令也类似。

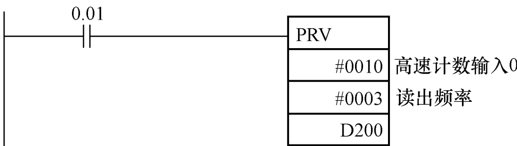


图 5-13　读脉冲频率程序

b）图 5-14a 所示为 2 个目标值比较程序。当 0.00 由 OFF 到 ON 时，执行 CTBL 指令，登录与起动比较。设定参数如图 5-14b 所示。从图知，本例设定为 2 个比较数。目标值分别为 1F4 及 3E8Hex。对应的中断程序为 1 与 2。中断程序的具体内容略。

提示：以上只是对指令的概要介绍，细节多与 PLC 型号、版本有关，可参阅有关说明书。建议在指令使用前最好先做些测试。

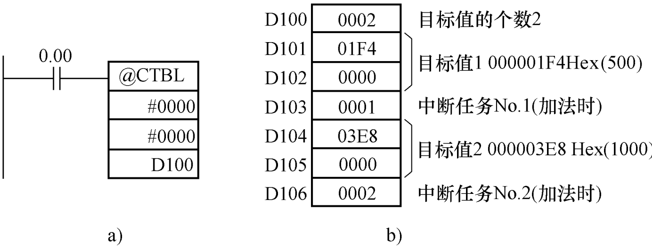


图 5-14　读脉冲频率程序

a）2 个目标值比较程序　b）参数设定

2. 高速计数单元输入

中、大型 PLC 处理脉冲量输入都是用高速计数单元。该单元自配有 CPU，可独立处理计数、比较以至于中断输出。然而，它又是通过总线与 PLC 的 CPU 挂接，故 CPU 也可向它读写数据，实现对高速计数的管理。

高速计数单元有不少类型，有单路的，仅处理一路高速计数；有多路的，可处理多路高速计数。

它的性能都比小型 PLC 高。如 C200H CT001 单元，其计数频率可高达 50kHz 或 75kHz，甚至于更高。最大计数范围为 -8388608 ~ +8388607。虽然只能处理一路输入脉冲，但可进行多达 16 个比较，每个比较最多能产生 16 路输出，其中有 8 路还可不通过 PLC 的 CPU，而由高速计数单元直接产生输出。

高速计数单元可用以进行脉冲计数，并能运用计数结果进行控制。

高速计数单元可安装在 CPU 机架或 I/O 扩展机架或远程从单元机架的 I/O 槽上，占用 10 个内部辅助继电器通道。具体占用的通道，依高速计数单元设定的机号确定，若机号为 Q，则所占的开始通道号 n 为（100 +10Q）。

C200H CT001 单元还要占 PLC 100 个数据存储器（DM），用以作参数设定。若机号为 Q，则所占的开始 DM 起始号 m 为（1000 +100Q）。

（1）输入。高速计数单元有两种输入：外部输入（即脉冲信号输入，以及复位输入或其它控制输入 Z、IN1、IN2）和内部输入（即 PLC 对高速计数器的输出，有各种工作命令、设定的参数等）。

1）外部计数输入，一般有多种形式，如：

脉冲与方向式：输入 A 为脉冲信号，可使计数值变化；输入 B 为方向信号，可决定计数值是增，还是减。

增减脉冲式：输入 A 的脉冲使计数器增计数，而输入 B 的脉冲则使计数器减计数。

两相输入式：A、B 为同频率的脉冲信号，并同接入计数器。若 A 相比 B 相超前 90°，则使计数器增计数；反之，B 相比 A 相超前 90°，则为减计数。这种输入方式与旋转编码器转动方向对应，而且还可使计数值倍增，可乘 2、乘 4，自然也可不倍增。倍增后可提高计数的分辨率。是否倍增，倍增多少，可用模块上的 DIP 开关设定。

Z，IN1、IN2 用于复位或进行计数控制。其作用依计数工作模式及模块上的 DIP 开关的设定确定。输入信号电压有 5V、12V、24V，可依信号发生器选定。选定后，要按要求接线。

2）内部输入指 PLC 向它输出的工作命令及设定参数。

工作命令来自被其占用的内部辅助继电器，有开始计数命令，用 n 通道 00 位，ON 计数（n 为所占用的内部辅助继电器的始通道）；

传送数据命令，用 n 通道 01 位，ON 传数；

外部输出使能命令，用 n 通道 02 位，ON 允许向外输出；

改变设置范围或预置值命令，用 n 通道 04 位，ON 进行改变；

读出错码命令，用 n 通道 05 位，ON 进行读错码，读出错码及出错地址存（n +5）通道；计数器内部复位命令，用 n 通道 06 位，ON 发内部复位信号；

强迫外部输出使能命令，用 n 通道 07 位，ON 允许内部强迫输出；

0～#7 输出位强迫输出命令，对应地用 n 通道 08～15 位，ON 时，可向对应位发输出 ON 命令；比较范围使能命令，用（n+1）通道，相应位 ON，则对应的比较范围的比较有效，等等。不过，计数模式不同，所用的命令也不完全相同，使用时要注意。

设定参数来自由它占用的 DM，有计数模式设定，用的 DM 号为 m，有 6 种设定与硬件对应；比较数的上、下限及输出有效位设定。它所用的 DM 号依计数模式而定。

另外，已设定的参数还可进行改变。替代的数存于内部辅助继电器有关通道。改变过程由执行改变命令实现。

（2）输出。有外部输出，作用于单元的输出端点，可直接实现对工作对象的控制；还有内部输出，把信号送向 PLC，由 PLC 再进行输出或处理。故它对于 PLC，就是从计数单元的输入。

1）外部输出，有 8 个，即#0～#7。高速计数单元有自己的处理单元，不仅可计数，而且当输出使能信号 ON 时，可依计数值与比较范围的比较结果直接在#0～#7 端产生相应的输出；或强制输出使能 ON 时，依强制输出信号情况，产生相应的输出。

直接输出可提高输出对输入的响应速度，提高工作精度。

2）内部输出，也有 8 个，分别输出到 PLC（n+9）通道的 08～15 位，而且在外部输出出现的同时，其信号也送向 PLC，送向上述通道的 00～07 位。这些信号送入 PLC 后，可通过 PLC 程序，间接用于控制。但它要在扫描周期 I/O 刷新阶段才能输出，故在时间上是有滞后的。

此外，计数器的当前值也要送 PLC，送到（n+6）、（n+7）两个通道。另外，还有很多标志，如进行计数标志，IN1、IN2 ON 标志，复位标志，出错情况标志，等等，也要送 PLC。PLC 可通过程序运用这些信息，对计数工作进行监控。

（3）计数模式。有六种计数模式：

1）线性计数模式可用三种输入模式的任何一种进行增减计数。从 0 开始计：增，最大可计到 +8388607；减，最小可计到 -8388608。可知，这里用的是带符号位的 24 位二进制的计数器，折合成十六进制为 6 位。

线性计数模式，其计数值超过上下限时，视为计数错误。

本模式可设定 16 组比较上下限，每次比较最多可产生 16 个输出。具体产生输出的情况是，当计数器的当前值大于某下限，而又小于该组的上限时，与输出有效位设定为 1 的对应的输出位，即转为 ON。

输出有效数设定及比较的上下限有 16 组（对应于可进行 16 个比较），每组占 5 个数据存储器，共占 80 个。这 80 个的编号为 DM（m+10）～DM（m+89）。从 DM（m+10）起，依次相邻的 5 个为一组。

每一组依次的第五个 DM 存输出有效位设定，第一、二个存低限数，第三、四个存高限数（上下限均为超过 4 位的十进制数，并带符号位）。

在输出有效位设定的 DM 中，哪一位设为 1，对应于高计数值落入这个组的上下限之间时，哪一个输出即变为 ON。

如某个组输出设定有效位：00 位为 1，02 位为 1，其它位为 0，上限为 1000，下限为 900。那么，计数器当前值为 900～1000 之间时，#0、#1 输出点可能 ON（还要看输出使能信号）。当然，其它组也可作别的设定。如另一组，输出设定：01、02 为 1，其它位为 0，

上限为 1200，下限为 1100。那么，计数器当前值为 1100～1200 之间时，输出点#1、#2 可能 ON。这两种情况下，#1 输出点均 ON。所以，输出点的 ON 状态，是各组 ON 输出的“或”，对于这一点，使用时要弄清楚。

当然，不进行 16 个比较，每个比较时不产生 16 个输出也是完全可以的，把不须产生输出的对应位设成 0 即可；而对不需进行比较的组，把输出位设定成全 0 也就可以了；或者也可把对应比较范围使能信号置成 0。

本模式下进行计数器复位的方法有 17 种，可组合使用 Z、IN1 及内部复位信号，也可单独使用其中的一个，由 DIP 开关的设定来确定。

附带要提及的是，尽管作了以上设定，但也可通过传送操作而改变若干设定值或计数器的当前值，可使用计数器实现对工作对象的控制更为灵活。

图 5-15 所示为计数时信号波形。

图 5-15　线性计数信号波形

注：a—正常输出　b—强制输出

从图 5-15 可知：

（a）计数开始于计数起始信号上升沿，而终止于它的下降沿。终止后，内部或外部输出状态保留。

（b）外部输出使能信号 OFF，外部输出被禁止，但内部输出不受影响。

（c）当强迫输出使能位 ON，即 n 通道 07 位 ON 时，可使强迫输出位的状态实现输出。强迫输出位的状态存于被高速计数器占用的内部辅助继电器 n 通道的 08～15 位（分别对应于#0～#7 输出点）。如强迫输出使能位 ON，这时若通道 08 位为 1，则#0 端输出 ON，若 08 位为 0，即#0OFF。其它位的情况，也与这类似。

（d）在计数起始信号有效期间，要产生正常的计数输出，强迫输出使能无效。

2）循环计数模式：计数器在 0 与预置的最大值之间计数。增计数超过最大值时，又从 0 开始计数；减计数小于 0 时，返回到最大值，再减计数。其特点是可以循环计数。其预置最大值的选用范围为 0～65535。其输出、复位、DM 设置、IR 中命令等，均与线性计数模式相同。

3）预置计数模式：它从预置值（最大可预置值为 8388607）开始计数（可增计数、可减计数，但主要是减计数），允许有 20 个预置值，而且可用传送命令一次更新其中的 6 个预置值。

输出仅 7 个。原 8 个可对外输出中的#0 输出计数一开始即转为 ON，计数到设定的某值时转为 OFF。#1、#2 的 ON、OFF 也有值限，也可设定；#3 输出不用；#4 ~ #7 用于产生与预置对应的输出。当计数器从预置值开始减计数，到零时，#4 ~ #7 依其输出是否有效的设定，将产生输出。

#4 ~ #7 输出 ON 的时间，由 DM（m + 11）设定，单位为 0.01s。若其内容设为 FFFF，则 ON 到下一个开始起动为止。

预置值及输出有效位设定有 20 种（#0 ~ #19），每种占两个 DM，从 DM（m + 14）直到 DM（m + 53）共 40 个。每隔两个，依次用于#0 ~ #19 的设定。

其中两个，如 DM（m + 14）及 DM（m + 15）为#0 比较设定，约定是：

+14 的 15 ~ 00 位及 + 15 的 11 ~ 00 位存预置值，最大为 8388607（转换成二进制存放）。

+15 的 15 ~ 12 位对应#4 ~ #7 输出有效。如 12 位为 1，则#4 输出有效，即从用这里设置的预置值减计数到零时，#4 输出 ON，并 ON 保持相应的时间；若 12 位为 0，#4 不产生输出。13 位对应于#5 输出，其余类推。

使#0 输出 OFF 的值（计数器当前值小于它时）存于 DM(m + 1) 及 DM(m + 2) 中，m + 2 中存高位。使#1 输出 ON 的上限值（计数器当前值小于和等于它时）存于 DM(m + 3) 及 DM(m + 4) 中，m + 4 中存高位。下限值（计数值当前值大于和等于它）存于 DM(m + 5) 及 DM(m + 6) 中，m + 6 中存高位。

使#2 输出 ON 的上限值（含义同#1）存于 DM(m + 7) 及 DM（m + 8）中，m + 8 中存高位。下限值（含义同#1）存于 DM(m + 9) 及 DM(m + 10) 中，m + 10 中存高位。

可见#0、#1、#2 三个输出点总是要使用（自然，也还可被输出使能 OFF 命令所禁止），但#4 ~ #7 四个输出点，则要依对输出有效位的设定情况而定。可以四个都用，也可只用其中的若干个。

预置计数模式时占用的内部辅助继电器的使用情况为：

通道 n：

00 位——起始计数命令位。它的上升沿，使计数器的当前值置为预置值，并使#0 输出点 ON。当选用预置命令 OFF 时，默认使用#0 的预置值及有效位的设置。另外，起始位 OFF 仍有保持计数作用。这点与前两个模式不同。故要停止计数须另用复位命令，不能只使用起始命令 OFF，如图 5-16 所示。

01 位——传数命令，ON 可改变 DM（m + 14）~ DM（m + 53）的预置值。

02 位——输出使能，ON 允许输出，OFF 时禁止输出。

03 位——未用。

04 位——选用预置值命令，在起始命令 ON 之前，04 位 ON，可选定 20 种中的一种预置值。这所选的 20 种的一种的编号存于（n + 1）通道的 15 ~ 08 位。

05 位——读出错命令位。

06 位——复位命令位，ON 时，使计数器复位，当前值转为零。预置计数模式可用五种

复位方式复位。可用内部复位方式，即用这里的复位命令位；也可用 IN1 控制信号；也可联合使用。图 5-16 为内部复位方式。

07 位——强制输出使能位。

08 ~ 15 位——对应于#0 ~ #7 输出点的强迫输出信号。

（n + 1）通道：

07 ~ 00 位未用

15 ~ 08 位在 0 到 19 间，设预置数。

（n + 2）通道：

传送数据源通道的起始号。

（n + 3）通道：

03 ~ 00 位指定源数据的存放区，其含义为 0—DM、1—I/O、2—LR、3—HR、4—AR。07 ~ 04 位，未用。

15 ~ 08 位，可设置成 1 ~ 6，相应地可传送 1 ~ 6 种预置值。

这里，（n + 2）、（n + 3）通道的使用与线性计数及循环计数模式时的使用相同。（n + 4）、（n + 5）、（n + 6）、（n + 7）的使用，基本上与线性及循环模式的相同。（n + 4）的08 ~ 15 位，原未用，现改为#0 ~ #7 输出标志，即对内输入。（n + 8）的 15 ~ 08 位，存当前预置值进行计数的号，即 20 种中的 1 种，#0 ~ #19。其它的虽占用，但未被使用。

图 5-16 所示为计数信号及计数值、输出等按时间对应的波形。

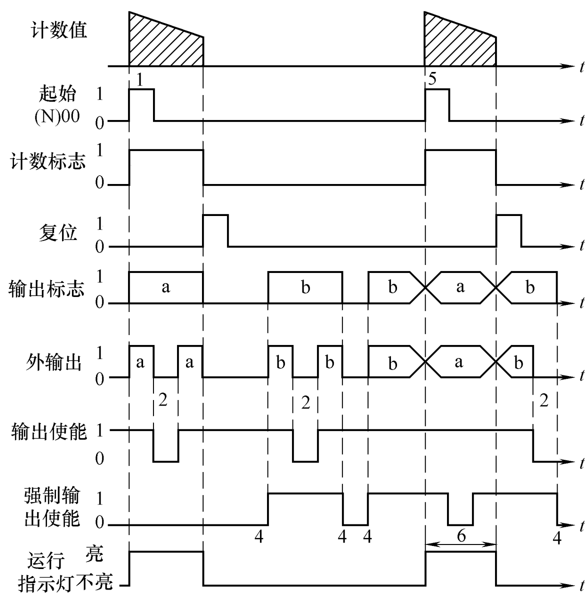


图 5-16　预置计数信号波形

从图 5-16 可知：

（a）起始位上升沿使计数器当前值变为预置值，并开始计数。

（b）当输出使能 OFF 时，输出被禁止。但对内输出不受影响（n + 4 通道的 08 ~ 15 位）。

（c）本例设内部复位有效，故复位位 ON 时，计数器复位，当前值变为 0，停止计数。

（d）强迫输出使能仅在不计数时才有效。

（e）强迫输出期间，若起始计数信号 ON，则回到正常计数的状态，即又从预置值起始计数，并使强迫输出停止工作。

（f）强迫输出时，复位信号 ON 后，先使外部输出 OFF，然后恢复为正常的强迫输出状态。

4）门控计数模式

门控计数模式有两种：一为正常的，另一为可累计的。

正常的门控计数模式：当控制信号 IN1 ON 时开始计数，OFF 时，停止计数，重新 ON，

又从零开始计数。

IN1 信号可为外部信号，也可为内部信号，由 DM 设定。

门控计数模式时，占用的 DM 仅 DM m 及 DM（m+1）两个。其情况是：DM m 的 11～08 位设成 4，设定为门控计数模式，其它位未用。DM（m+1）的 03～00 位，设成 0 时外部 IN1、IN2 有效，设成 1 时内部 IN1、IN2 有效。内部 IN1、IN2 用的内部辅助继电器，见后。07～04 位未用，设成 0。11～08 位，设定门控计数的两种类型：0 时为正常的门控计数模式；1 时为累计门控计数模式，15～12 位未用，设成 0。其它的 DM 位均未用。

对内部辅助继电器也仅用其 5 个通道。其情况是：

通道 n：

00 位～02 位——未用

03 位——内部控制信号 IN1

04 位——内部控制信号 IN2

05 位——读出错命令位

15～06 位——未用

通道（n+4）：

00 位——计数标志

01 位～02 位——未用

03 位～04 位——IN1、IN2 标志

05 位——出错标志

06 位——计数值溢出，即计数值在 -8388608～+8388607 范围之外

15 位～07 位——未用

通道（n+5）：

记录出错信息

通道（n+6）、（n+7）：

存计数器当前值

5）闩锁计数模式

它的计数用 IN1 上升沿起动计数，起动计数后计数不停地进行，再送入 IN1，将使计数又从零开始。

IN2 控制闩锁，其上升沿起作用。上升沿进行闩锁，即在反映计数值的内部辅助继电器中被闩锁住，保持不变，而计数器计数不受影响。而再送入 IN2，其上升沿又使当时计数值读入内部辅助继电器中，并又被闩锁住。

闩锁计数模式时 DM、IR 的使用情况，与门控的类似，只是计数模式设成 5。

6）采样计数模式

采样计数模式是在给定的时间内进行计数。给定时间也称为采样时间，此时间值存于内部辅助继电器的（n+1）通道，并由 n 通道的 01 位时间设置信号使能。

计数的起动控制用 IN1 的上升沿。在起动计数前，采样时间可由时间设置信号使能，送入新值。自然，这时（n+1）通道的内容要作改变。

在采样时间相同时，其计数值代表速度，故这种计数模式可用于检测脉冲频率。

采样模式时的 DM、IR 的其它使用同门控制时的情况，只是计数模式设成 6。

应该指出的是，4 ~6 三种模式都没有外部输出，只能通过 PLC 的输出点产生输出，这在响应速度上是要受扫描周期影响的，故出于快速响应的要求，一般不用这三种模式。

提示：以上只是一种 PLC 的一种高速计数模块。其实，每种中大型 PLC 多都有几种不同规格与特性的高速计数模块，其细节与这里介绍的不可能完全相同。要准确使用可参阅有关说明书。

5. 2. 3　脉冲信号输出

中、大型 PLC 输出脉冲，使用位置控制单元或运动单元。这些单元有自己的 CPU、内存及输出与输入点。通过与 PLC CPU 交换数据，或预先设定，可确定通过哪个输出口发送脉冲、送多少脉冲及脉冲的频率多大等问题。确定之后，这些单元可自行工作。所以，这类 PLC 就没有自身的脉冲输出口及脉冲输出指令。有关这些单元将在本章第 5. 6 节作简要介绍。

小型 PLC 可指定多个输出点输出脉冲，如 CP1H 机有 2 个脉冲输出口，如果为 Y 型 PLC 则有 4 个输出口。此外，还有 2 个可调制脉宽输出口，如图 5-17 所示。表 5-4 所示为脉冲输出使用的辅助继电器。

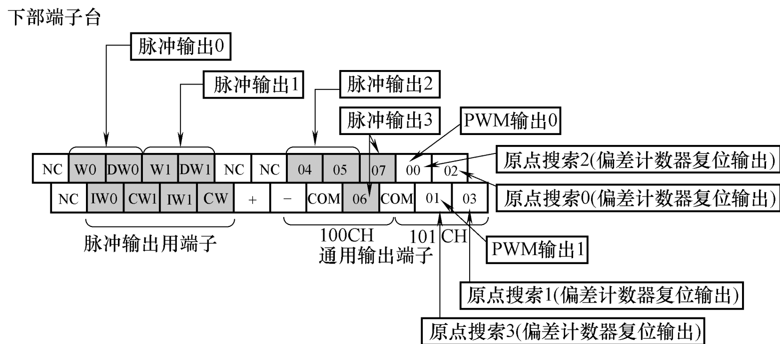


图 5-17　CP1H Y 型 PLC 脉冲输出点分布

表 5-4　CP1H 机脉冲输出使用辅助继电器

内　容		脉冲输出 0	脉冲输出 1	脉冲输出 2	脉冲输出 3
当前值保存区域 80000000 ~ 7FFFFFFF Hex (−2147483648 ~ 2147483647 脉冲)	保存高位 4 位	A277CH	A279CH	A323CH	A325CH
	保存低位 4 位	A276CH	A278CH	A322CH	A324CH
脉冲输出复位标志 清除脉冲输出当前值区域	0：不清除 1：清除	A540.00	A541.00	A542.00	A543.00
CW 界限输入信号 原点搜索中使用的 CW 界限输入信号	来自外部的输入为 ON 时，ON	A540.08	A541.08	A542.08	A543.08
CCW 界限输入信号 原点搜索中使用的 CCW 界限输入信号	来自外部的输入为 ON 时，ON	A540.09	A541.09	A542.09	A543.09

（续）

内　　容		脉冲输出0	脉冲输出1	脉冲输出2	脉冲输出3
定位结束信号 原点搜索中使用的定位结束信号	来自外部的输入为ON时,ON	A540.10	A541.10	A542.10	A543.10
脉冲输出状态标志 通过ACC/PLS2指令使脉冲输出中输出频率发生阶段性变化,并在加减速中为ON	0:恒速中 1:加减速中	A280.00	A281.00	A326.00	A327.00
溢出/下溢标志 计数值为溢出或下溢时为ON	0:正常 1:发生中	A280.01	A281.01	A326.01	A327.01
脉冲输出量设定标志 通过PLUS指令,设定脉冲量时为ON	0:无设定 1:有设定	A280.02	A281.02	A326.02	A327.02
脉冲输出结束标志 通过PULS/PLS2指令设定的脉冲量结束输出时,为ON	0:输出未结束 1:输出结束	A280.03	A281.03	A326.03	A327.03
脉冲输出中标志 脉冲输出中时为ON	0:停止中 1:输出中	A280.04	A281.04	A326.04	A327.04
无原点标志 原点未确定时为ON	0:原点确认状态 1:原点未确认状态	A280.05	A281.05	A326.05	A327.05
原点停止标志 脉冲输出的当前值与原点(0)一致时为ON	0:位于原点以外的停止中 1:位于原点的停止中	A280.06	A281.06	A326.06	A327.06
脉冲输出停止异常标志 原点搜索功能中,脉冲输出中发生异常时为ON	0:无异常 1:异常发生	A280.07	A281.07	A326.07	A327.07
停止异常代码 发生脉冲输出停止异常时,该异常代码被保存		A444CH	A445CH	A438CH	A439CH

CP1H机可实现的脉冲输出功能有：根据驱动器的特点，选择CW/CCW脉冲输出或脉冲+方向输出；在绝对坐标系中，可自动选择CW/CCW的方向；可加速及减速输出脉冲；可变更目标位置、目标速度、加速比率、减速比率；可在速度控制中变更定位；可在加速或减速中变更目标速度、加减速比率；可发出可变占空比脉冲输出信号（PWM）。表5-5所示为CP1H机脉冲输出功能。

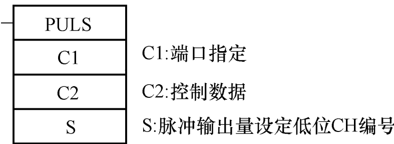
表 5-5　CP1H 机脉冲输出功能

目　　的	使 用 功 能	内　　容
用定位用脉冲输出，输出到脉冲串输入的电动机驱机器，进行简单定位	脉冲输出功能 ・无加减速的单相脉冲输出(通过 SPED 指令) ・有加减速的单相脉冲输出(加速、减速比率相同的梯形)(通过 ACC 指令) ・有梯形加减速的单相脉冲输出(加速、减速比率不同的梯形，有起动频率)(通过 PLS2 指令)	X/XA 型的情况下，可将内置输出触点作为脉冲输出 0、1、2、3 使用 Y 型的情况下目标频率： X/XA 型：1Hz～100kHz(脉冲输出 0、1 的情况下)、1Hz～30kHz(脉冲输出 2、3 的情况下)、Y 型：1Hz～1MHz(脉冲输出 0、1 的情况下)、1Hz～30kHz(脉冲输出 2、3 的情况下) 占空比：50% 脉冲输出的种类可为 CW/CCW、脉冲 + 方向控制中的任一种。但是，脉冲输出 0、1 的情况下，限于同一方式 注：脉冲输出当前值被保存到特殊辅助继电器区域
原点搜索/复位	原点决定(原点搜索、原点复位)功能	在脉冲输出时，可进行原点搜索/复位 ・原点搜索：通过 PLC 系统设定为“使用原点搜索”，并在设定各种参数的状态下、执行 ORG(原点搜索)指令时，开始原点搜索，将原点确定为原点附近输入信号及原点输入信号。此时，脉冲输出当前值的坐标自动变为绝对坐标 ・原点复位：通过 PLC 系统设定“原点复位起动速度”等的参数的状态下，执行 ORG(原点搜索)指令时，复位到已确定的原点
希望在定位中更改目标位置(多重起动导致的紧急回避动作等)	通过 PLS2 指令定位	可在脉冲输出(PLS2)指令进行的定位中，通过执行其它的脉冲输出(PLS2)，可变更目标位置、目标速度、及加速比率、减速比率
在速度控制中，希望将速度变更为近似折线	通过 ACC 指令(连续)的加速比率或减速比率的变更	可在 ACC 指令(连续)进行的速度控制中，变更加速比率或减速比率，执行 ACC 指令(连续)
在位置控制中，希望将速度变更为近似折线	通过 ACC 指令(单独)或 PLS2 指令的加速比或减速比的变更	可在 ACC 指令(单独)或 PLS2 指令进行的位置控制中，变更加速比率或减速比率，执行 ACC 指令(单独)或 PLS2 指令
希望进行恒定距离进给	通过 SPED 指令(连续)或 ACC 指令(连续) + PLS2 指令的定位	在 SPED 指令(连续)或 ACC 指令(连续)进行的速度控制中，通过执行脉冲输出(PLS2)，切换到定位。可在输出恒定的脉冲量后停止
原点确定后，希望进行与当前位置及目标位置的方向关系无关，绝对位置坐标下的简单定位	通过绝对坐标系下的绝对脉冲指定的定位方向自动选择功能	在绝对坐标系下动作时(通过原点确定状态或 INI 指令执行当前值变更)，根据指令指定的脉冲输出量与脉冲输出当前值相比为正或为负，脉冲输出指令执行时 CW/CCW 的方向被自动选择
希望进行三角控制	通过 ACC 指令(单独)或 PLS2 指令进行定位	定位(ACC 指令(单独)或 PLS2 指令执行)中，加速及减速时必要的脉冲输出量(达到目标频率的时间 × 目标频率)超过设定的目标脉冲输出量的情况下，进行三角控制(无恒定速度时间的梯形控制)
希望使用可变占空比输出，进行温度的时间比例控制	模拟输入 + 可变占空比输出脉冲输出功能(PWM)	通过执行 PWM 指令，可将内置输出触点作为 PWM 输出 0、1 使用

实现这些功能，要使用相应的脉冲输出指令。如 CMP2A，除了第 4 章已介绍 PMW 指令外，还有设置脉冲 PULS 指令、频率输出 SPED 指令、加速控制 ACC 指令、同步控制 SYNC 指令。CJ1M 机、CP1H 机还有功能更强的 PLS2 指令。以下对这些指令作简要介绍。只是要注意，这些指令的细节可能随着机型、版本的不同而有所不同。建议使用前可先做些测试。

1. PULS 指令

本指令用以设定输出的脉冲数。其梯形图格式如下：



这里，C1 用以指定输出口。C2 为控制数据。S、S+1 输出脉冲数设定。这些指令参数含义及选用见表 5-6。

表 5-6　PULS 指令参数的含义及其选用

操作数			内容
C1	端口指定		#0000 Hex:脉冲输出 0 #0001 Hex:脉冲输出 1 #0002 Hex:脉冲输出 2 #0003 Hex:脉冲输出 3
C2	控制数据		#0000 Hex:相对脉冲指定 #0001 Hex:绝对脉冲指定
S	脉冲输出量设定低位 CH	S CH(低位 4 位)	・相对脉冲指定时： 00000000～7FFFFFFF Hex(0～2147489647) ・绝对脉冲指定时： 80000000～7FFFFFFF Hex (－2147489648～2147489647)
		S+1CH(高位 4 位)	

实际的输出脉冲数与相对或绝对脉冲有关。相对脉冲指定时为

移动脉冲量=脉冲输出量设定值

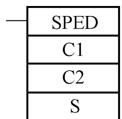
绝对脉冲指定时为

移动脉冲量=脉冲输出量设定值－当前值

脉冲输出中不能进行脉冲输出量的再设定，这时执行 PULS 指令时为出错。因此本指令只能采用输入微分执行。

2. SPED 指令

本指令用以指定要输出的脉冲频率，并起动脉冲输出。其梯形图格式如下：



这里，C1 为指定端口。C2 为指定输出模式。S、S+1 指定目标频率，如设为 0，指定脉冲输出停止。这些指令参数含义及选用见表 5-7。

表 5-7　SPED 指令参数的含义及其选用

操作数			内容
C1	端口指定		#0000 Hex:脉冲输出 0 #0001 Hex:脉冲输出 1 #0002 Hex:脉冲输出 2 #0003 Hex:脉冲输出 3
C2	输出模式	0～3 位	模式指定 0 Hex:连续模式 1 Hex:单独模式
		4～7 位	方向指定 0 Hex:CW 1 Hex:CCW
		8～11 位	脉冲输出方式 0 Hex:CW/CCW 输出 1 Hex:脉冲＋方向输出
		12～15 位	未使用 0 Hex 固定
S	目标频率低位 CH	S CH(低位 4 位)	2CH 占用 将输出频率以 1Hz 单位指定 · X/XA 型: 脉冲输出 0、1:00000000～000186A0 Hex(0～100kHz) 脉冲输出 2、3:00000000～00007530 Hex(0～30kHz) · Y 型: 脉冲输出 0、1:00000000～000F4240 Hex(0～1MHz) 脉冲输出 2、3:00000000～00007530 Hex(0～30kHz)
		S＋1CH(高位 4 位)	

SPED 指令的功能是，由 C1 指定的端口、由 C2 指定的模式和由 S、S＋1 指定的目标频率输出脉冲。执行一次 SPED 指令，即输出脉冲。因此基本上在输入微分型（带@）或 1 周期 ON 的输入条件下使用。

独立模式时，事先由 PULS 指令设定脉冲量。连续模式时，脉冲输出一直进行到执行停止脉冲输出为止。表 5-8 及 5-9 所示分别为连续与独立模式输出的各种应用。

表 5-8　连续模式输出

操作	动作内容	使用示例	频率的变化	说明	使用步骤 指令语句
脉冲输出开始	速度指定输出	使速度(频率)变为阶跃式时	脉冲频率 目标频率 时间 SPED命令执行	进行指定频率的脉冲输出	SPED(连续)
设定变更	阶跃式速度变更	希望在运行中变更速度时	脉冲频率 目标频率 当前的频率 时间 SPED命令执行	将脉冲输出中的频率变更为阶跃式(上方或下方)	SPED(连续) ↓ SPED(连续)

（续）

操作	动作内容	使用示例	频率的变化	说明	使用步骤 指令语句
脉冲输出停止	脉冲输出停止	即刻停止	脉冲频率 当前的频率 时间 INI命令执行	即刻停止脉冲输出	SPED（连续） ↓ INI
	脉冲输出停止	即刻停止	脉冲频率 当前的频率 时间 SPED命令执行	即刻停止脉冲输出	SPED（连续） ↓ SPED（连续、目标频率0Hz）

表 5-9　独立模式输出

操作	动作内容	使用示例	频率的变化	说明	使用步骤 指令语句
脉冲输出开始	速度指定输出	进行无加减速的定位时	脉冲频率 指定脉冲量（由PULS命令指定） 目标频率 时间 执行SPED命令 输出指定脉冲量的停止	按照指定脉冲输出的频率开始，输出指定脉冲量时，使其即刻停止 注：不可变更定位（脉冲输出）中的目标位置（脉冲量）	PULS ↓ SPED（独立）
设定变更	阶跃式速度变更	希望在运行中将速度变更为阶跃式时	指定隙冲量（由PULS命令指定） 脉冲频率 （根据PULS命令脉冲量不变） 变更后目标频率 目标频率 时间 执行SPED命令(独立) 执行SPED命令(独立)变更目标频率（目标位置没有变化）	定位中，执行SPED指令，将脉冲输出中的频率变更为阶跃式（上方或下方）。该情况下，目标位置（脉冲量）不变	PULS ↓ SPED（独立） ↓ SPED（独立）
脉冲输出停止	脉冲输出停止（脉冲输出量不保持）	即刻停止	脉冲频率 目标频率 时间 执行SPED命令 执行INI命令	即刻停止脉冲输出，此时，当前的脉冲输出量被清除	PULS ↓ SPED（独立） ↓ INI
	脉冲输出停止（脉冲输出量不保持）	即刻停止	脉冲频率 目标频率 时间 执行SPED命令 执行SPED命令	即刻停止脉冲输出，此时，当前的脉冲输出量被清除	PULS ↓ SPED（独立） ↓ SPED（独立、目标频率0Hz）

图 5-18a 所示为 SPED 指令的一个应用实例。执行本程序，当 0.00 由 OFF→ON 时，通过 PULS 指令由相对脉冲指定将脉冲输出 0 的脉冲输出量设定为 5000 脉冲。同时通过 SPED 指令由 CW/CCW 输出、CW 单独模式开始输出目标频率 500Hz 的脉冲（设定值如图 5-18b 所示）。执行效果如图 5-18c 所示。

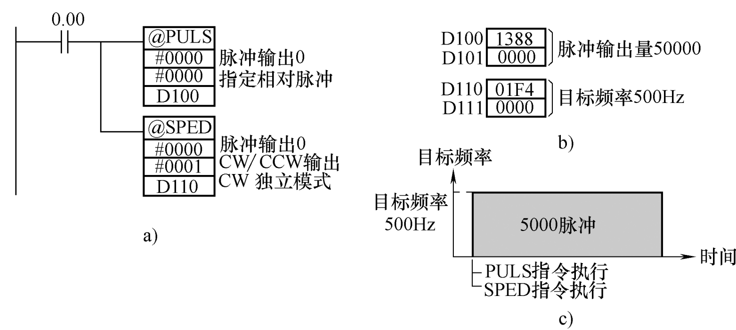
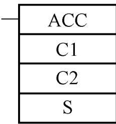


图 5-18　SPED 指令应用实例

a）应用实例　b）设定值　c）执行效果

3. ACC 指令

本指令用以指定输出的脉冲频率按梯形规律增加与减少，并起动输出。其梯形图格式如下：



这里，C1 用以输出端口指定。C2 为操作数。S 用以指定加、减速率。这些指令参数含义及选用见表 5-10。

表 5-10　ACC 指令参数的含义及其选用

操作数			内容
C1	端口指定		#0000 Hex:脉冲输出 0 #0001 Hex:脉冲输出 1 #0002 Hex:脉冲输出 2 #0003 Hex:脉冲输出 3
C2	模式指定	0～3 位	模式指定 0 Hex:连续模式 1 Hex:单独模式
		4～7 位	方向指定 0 Hex:CW 1 Hex:CCW
		8～11 位	脉冲输出方式 0 Hex:CW/CCW 输出 1 Hex:脉冲 + 方向输出
		12～15 位	未使用 0 Hex 固定

（续）

操作数			内容
S	设定表低位 CH	S CH	加减速比率 0001 ~ FFFF Hex(1 ~ 65535 Hz) 将每个脉冲控制周期(4ms)的频率增减量以 1 Hz 单位设定
		S+1CH(低位 4 位)	目标频率 ・X/XA 型: 脉冲输出 0、1:00000000 ~ 000186A0 Hex(0 ~ 100kHz) 脉冲输出 2、3:00000000 ~ 00007530 Hex(0 ~ 30kHz) ・Y 型: 脉冲输出 0、1:00000000 ~ 000F4240 Hex(0 ~ 1MHz) 脉冲输出 2、3:00000000 ~ 00007530 Hex(0 ~ 30kHz) 将加减速后的频率以 1 Hz 单位指定
		S+2CH(高位 4 位)	

ACC 指令的功能是由 C1 指定端口由 C2 指定方式由 S+1 与 S+2 指定目标频率和 S 指令加减速比率输出脉冲。在到达目标频率之前，在每个脉冲控制周期（4ms）中，按照加减速比率，进行频率的加减速。图 5-19 所示的为输出脉冲频率加、减的过程。

执行一次 ACC 指令，即可按指令输出脉冲。所以，本指令都是微分执行。

独立模式时，输出脉冲数由 PULS 指令设定。完成脉冲数输出后自动停止。在连续模式中，脉冲输出一直进行到执行停止脉冲输出为止。

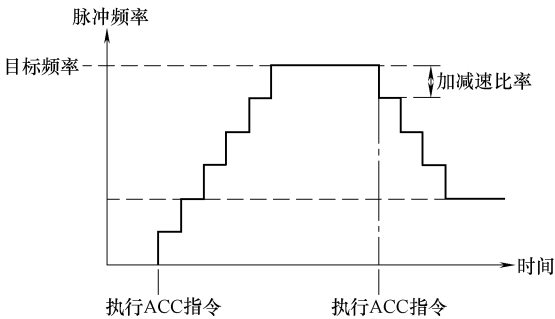


图 5-19　输出脉冲频率加、减的过程

图 5-20a 为 ACC 指令连续模式输出应用程序实例，设定如图 5-20b 所示。执行本程序，当 0.00 由 OFF→ON 时，从脉冲输出端口 0 ，用 CW/CCW 输出、CW、连续模式开始进行加减速比率 20Hz、目标频率 500Hz 的脉冲输出。之后 0.01 由 OFF→ON 时，再一次通过 ACC 指令变更为加减速比率 10Hz、目标频率 1000Hz。图 5-20c 所示为程序运行结果示意。

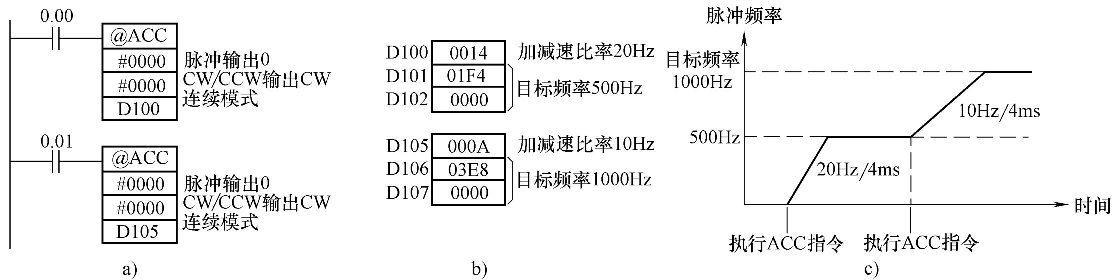
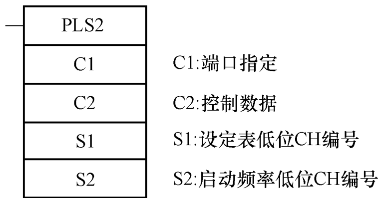


图 5-20　ACC 指令应用实例

a）应用实例　b）设定值　c）运行结果

4. PLS2 指令

本指令用以指定要输出的脉冲数。其梯形图格式如下：



这里，C1 用以端口指定。C2 为控制数据。S2 用以指定起动时的脉冲频率，S1 为加速比率，S1 +1 为减速比率，一般可在 1 ~ 65535 之间选定；S1 +2、S1 +3 为目标频率，一般可在 1 ~ 100000 之间选定。S1 +4、S1 +5 为输出脉冲数。这些指令参数含义及选用见表 5-11。

表 5-11　PLS2 指令参数的含义及选用

操作数			内容	
C1	端口指定		#0000 Hex:脉冲输出 0 #0001 Hex:脉冲输出 1 #0002 Hex:脉冲输出 2 #0003 Hex:脉冲输出 3	
C2	控制数据	0 ~ 3 位	模式指定 0 Hex:相对脉冲指定 1 Hex:绝对脉冲指定	
		4 ~ 7 位	方向指定 0 Hex:CW 1 Hex:CCW	
		8 ~ 11 位	脉冲输出方式 0 Hex:CW/CCW 输出 1 Hex:脉冲 + 方向输出	
		9 ~ 15 位	未使用 0 Hex 固定	
S1	设定表低位 CH	S CH	加减速比率 0001 ~ FFFF Hex(1 ~ 65535Hz)	将每个脉冲控制周期(4ms)的频率以 1Hz 单位设定
		S1 +1CH	减速比率 0001 ~ FFFF Hex(1 ~ 65535Hz)	
		S1 +2CH(低位 4 位)	目标频率 · X/XA 型: 脉冲输出 0、1:00000000 ~ 000186A0 Hex(0 ~ 100kHz) 脉冲输出 2、3:00000000 ~ 00007530 Hex(0 ~ 30kHz) · Y 型: 脉冲输出 0、1:00000000 ~ 000F4240 Hex(0 ~ 1MHz) 脉冲输出 2、3:00000000 ~ 00007530 Hex(0 ~ 30kHz) 将加减速后的频率以 1Hz 单位指定	
		S1 +3CH(高位 4 位)		

（续）

操　作　数			内　　容
S1	设定表低位 CH	S1 +4CH（低位 4 位）	脉冲输出量设定 ・相对脉冲指定时： 00000000 ~7FFFFFFF Hex（0 ~2,147,489,647） ・绝对脉冲指定时： 80000000 ~7FFFFFFF Hex（ -2,147,489,648 ~2,147,489,647）
		S1 +5CH（高位 4 位）	
S2	起动频率下位 CH	S2CH （低位 4 位）	将起动时的频率以 1Hz 单位指定 ・X/XA 型： 脉冲输出 0、1：00000000 ~000186A0 Hex（0 ~100kHz） 脉冲输出 2、3：00000000 ~00007530 Hex（0 ~30kHz） ・Y 型： 脉冲输出 0、1：00000000 ~000F4240 Hex（0 ~1MHz） 脉冲输出 2、3：00000000 ~00007530 Hex（0 ~30kHz）
		S2 +1CH（高位 4 位）	

实际的输出脉冲数与相对或绝对脉冲有关。相对脉冲指定时为

$$移动脉冲量 = 脉冲输出量设定值$$

绝对脉冲指定时为

$$移动脉冲量 = 脉冲输出量设定值 - 当前值$$

PLS2 指令执行功能如图 5-21 所示。执行时，由 C1 指定端口，由 C2 指定控制数据，由 S2 指定起动频率、输出脉冲（图中①）。在每个脉冲控制周期（4ms）中，根据指定的加速比率，在达到目标频率之前，使频率增加（图中②）。达到目标频率之后停止加速，以等速继续脉冲的输出（图中③）。到达指定的脉冲输出量和减速比率中所计算得到的减速点（使频率减少的定时）之后，在每个脉冲控制周期（4ms）中对频率进行减少，当到达起动频率时，停止输出（图中④）。

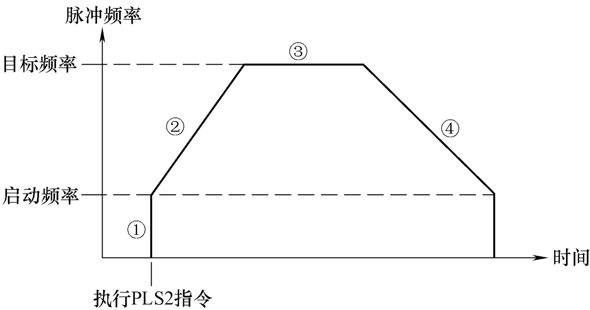


图 5-21　PLS2 指令执行效果

执行一次 PLS2 指令时，由指定条件开始输出脉冲，所以本指令为微分执行。表 5-12 及表 5-13 所示分别为独立与连续模式输出的各种应用。

图 5-22a 为 ACC 指令一个应用实例，设定如图 5-22b 所示。执行本程序，当 0.00 由 OFF→ON 时，从脉冲输出口 0 开始（脉冲数为 10000）输出脉冲。起动频率为 200Hz。加速比率为 500Hz/4ms。加速到目标频率 50kHz 为止。之后从减速点，以 250Hz/4ms 的减速比率进行减速，当减速到起动频率的 200Hz 时，停止脉冲输出。图 5-22c 为程序执行的效果。

表 **5-12** 独立模式输出

操作	动作内容	使用例	频率的变化	说明	使用顺序 指令语
脉冲输出开始	详细的梯形控制	在决定梯形加减速的位置时(加速比率和减速比率为个别,有启动速度) 能够变更定位中的脉冲量	脉冲频率 指定脉冲量 目标频率 加速频率 减速比率 加减频率 停止比率 时间 执行PLS2指令 停止输出 执行目标频率 减速点	以一定的比率对频率进行加速,以一定的比率对频率进行减速,输出指定脉冲量时,停止加减速 1) 注:能够变更定位(脉冲输出)中的目标位置(脉冲量)	PLS2
设定变更	带斜率速度变更(加速比率≠减速比率)变更	定位中的目标速度(频率)的变更(加速比率和减速比率为个别)	脉冲频率 指定脉冲量 (用PLS2指令确定) 根据PLS2命令的脉冲量不变 变更后目标频率 目标频率 加减速比率 时间 执行PLS2指令 执行PLS2指令 变更目标频率 (目标位置没有变化,但是加减速比率被变更)	在定位中执行 PLS2 指令,能够对加速比率、减速比率、目标频率进行变更 注:在有意识不改变目标位置时,作为 PLS2 指令的操作数,有必要指定在绝对指定中和原来相同的位置	PLS2 ↓ PLS2 PULS2 ↓ ACC(独立)↓ PLS2
	目标位置变更	变更定位中的目标位置(多重启动)	脉冲频率 根据PLS2指令的脉冲量变更 指定脉冲量 目标频率 加减速比率 时间 执行PLS2命令 执行PLS2指令 目标位置变更 (目标位置和加减速比率没有变更)	在定位中执行 PLS2 指令,能够变更目标位置(脉冲量)、加速比率、减速比率、目标频率 注:在变更后不能确保等速域时为出错,无视执行,继续原来的动作	PLS2 ↓ PLS2 PULS ↓ ACC(独立)↓ PLS2

（续）

操　作	动作内容	使用例	频率的变化	说　明	使用顺序 指令语
设定变更	目标位置+带斜率速度变更	变更(多重启动)定位中的目标位置、目标速度(频率)(多重启动)	脉冲频率 根据PLS2指令的脉冲量变更 指定脉冲量 变更后目标频率 目标频率 加减速比率 时间 执行PLS2指令 根据PLS2指令的脉冲量变更	在定位中执行PLS2指令，能够变更目标位置(脉冲量)、加速比率、减速比率、目标频率 注：在变更后不能确保等速域时为出错，无视执行，继续原来的动作	PLS2 ↓ PLS2 PULS ↓ ACC(独立)↓ PLS2
		定位中的加减速的变更(多重启动)	脉冲频率 根据PLS2指令的脉冲量变更 加减速比率4 变更后目标频率 加减速比率3 目标频率 加减速比率2 加减速比率1 时间 执行PLS2指令 执行PLS2指令 执行PLS2指令 执行PLS2指令	在定位中执行PLS2指令，能够变更加速比率、减速比率	PLS2 ↓ PLS2
	方向变更	定位中的方向变更	脉冲频率 指令脉冲量 根据指定的减速比率变更方向 目标频率 根据PLS2命令的脉冲量变更 时间 执行PLS2指令 执行PLS2指令	在通过绝对脉冲指定的定位中执行PLS2指令，能够指定绝对位置，变更方向	PLS2 ↓ PLS2 PULS ↓ ACC(独立) PLS2

（续）

操　　作	动作内容	使用例	频率的变化	说　　明	使用顺序 指令语
脉冲输出停止	脉冲输出停止（脉冲输出量非保持）	立即停止	脉冲频率 目标频率 时间 执行PLS2指令 执行INI指令	立即停止脉冲输出，这时清除当前的脉冲输出量	PLS2 ↓ INI
	带斜率脉冲输出停止（脉冲输出量非保持）	减速停止	脉冲频率 目标频率 加速比率 目标频率=0 时间 执行ACC指令 执行ACC指令	减速并停止脉冲输出	PLS2 ↓ ACC（独立、目标频率0Hz）

表 5-13　连续模式输出

使 用 例	频率的变化	说　明	使用顺序 指令语言
速度控制中的向定量传送定位的变更	输出由PLS2命令所指定的脉冲量(能指定绝对脉冲或相对脉冲) 脉冲频率 目标频率 时间 执行ACC指令 执行PLS2指令	在由 ACC 指令来进行速度的控制中，执行 PLS2 指令，变更为定位	ACC（连续） ↓ PLS2
中断定量传送	脉冲频率 目标频率 时间 执行ACC指令 通过以下的设定执行PLS2指令 −脉冲量=到停止为止的脉冲量 −设定绝对脉冲 −目标频率=当前的频率 −加速比率 −减速比率 =目标减速比率		

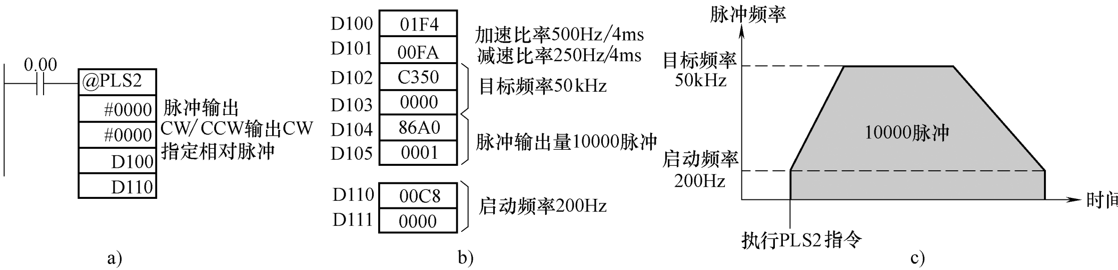
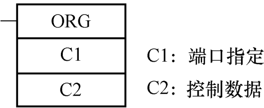


图 5-22　PLS2 应用实例

a）应用实例　b）设定值　c）执行效果

5. ORG 指令

本指令用以进行原点搜索或原点复位。其梯形图格式如下：



这里，C1 用以端口指定。0000 Hex：脉冲输出 0。0001 Hex：脉冲输出 1；0002 Hex：脉冲输出 2（仅 CP1H）；0003 Hex：脉冲输出 3（仅 CP1H），0020 Hex：变频器定位 0（仅 CP1L）。

C2 为控制数据。0 ~ 3 固定为 0，4 ~ 7 也固定为 0；8 到 11：0 为 CW/CCW 输出，1 为脉冲加方向输出；9 到 15：0 为原点搜索，1 为原点复位。

原点搜索是通过输出脉冲驱动运动部件，使用“原点附近”信号的输入、“原点信号”输入来确定原点。原点复位是从当前位置向确定的原点移动。

ORG 指令主要用于绝对坐标系，详细内容请参见有关手册。

6. PRV 指令补充说明

PRV 指令在脉冲输入、输出中都很有用。在脉冲输出时，可用以读取输出状态。这时，应把它的 C2 设定为 0001Hex。至于读取哪个口状态，可用 C1 设定输出口（0000 Hex：脉冲输出口 0、0001 Hex：脉冲输出口 1）。所读的状态数据存于 D 字中。其中第 4 位为 1，表示脉冲正在输出，0 表示脉冲输出已完成（停止）。用它控制脉冲的正确输出是很方便的。

5.2.4　脉冲信号执行

脉冲信号执行与系统的功能与配置有关。如果用脉冲频率大小作为控制输出，则可用 FVC（频率到电压）变换。变换为电压信号后，再经功率放大，即可实现各种控制了；如果用脉冲宽度作为控制输出，则可直接或稍作滤波，再经功率放大也可实现各种控制了。脉宽控制输出还可用以控制舵机、机械手。如果用脉冲量及频率去控制运动机构运动，就要用相应的执行机构。有两种机构：一用步进电动机；另一用伺服电动机，如图 5-23 所示。

以下仅对步进电动机与伺服电动机作简要介绍。

1. 步进电动机

步进电动机是将脉冲信号转换成角位移或线位移的控制电动机。它用专用的驱动电源，提供一系列有规律的电脉冲信号。其角位移与脉冲数成正比，电动机转速与脉冲频率成正比，而且转速和转向与各相绕组的通电方式有关。

（1）原理。图 5-24 是三相反应式步进电动机的结构示意图。

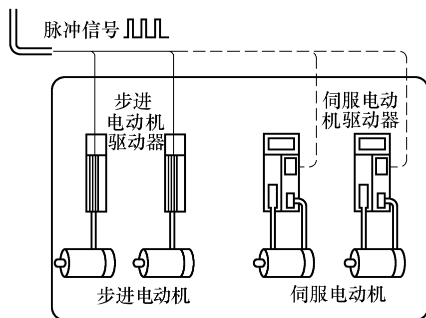


图 5-23　脉冲信号执行机构

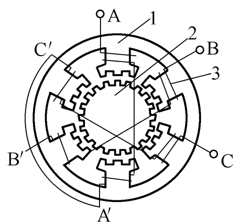


图 5-24　三相反应式步进电动机的结构示意图

1—定子　2—转子　3—定子绕组

从图 5-24 知，电动机的定子上有 6 个均布的磁极，其夹角是 60°。各磁极上套有绕组，连成 A、B、C 三相绕组。转子上均布 40 个小齿。每个齿的齿距为 $\theta_E = 360°/40 = 9°$。每个定子磁极也有小齿，但各相有 1/3 齿距（3°）的错位。若 A 相磁极齿和转子齿对齐，则 B 相向前错位 1/3 齿距，C 相向后错位 1/3 齿距。若 B 相、C 磁极齿和转子齿对齐，其它两相情况也类似。而且，哪个相磁极齿和转子齿对齐，哪个相磁阻就最小。

这样，当 A 相对齐，给 B 相通电，B 相绕组产生定子磁场，其磁力线穿越 B 相磁极，力图使转子转动，直到转过 3°、使 B 磁极齿与转子齿对齐。接着，如停止对 B 相绕组通电，而改为对 C 相绕组通电，同理，转子按顺时针方向再转过 3°。依次类推，当三相绕组按 A→B→C→A 顺序循环通电时，转子会按顺时针方向，以每个通电脉冲转动 3°。若改变通电顺序，按 A→C→B→A 顺序循环通电，则转子就按逆时针方向转动。

这样通电，称为单三拍方式，步距角 θ_b 为 3°。也可双三拍方式，即 AB→BC→CA→AB 顺序循环通电。也还可单、双六拍运行，即按 A→AB→B→BC→C→CA→A 顺序循环通电。六拍运行的步距角将减小一半。步距角可按下式计算：

$$\theta_b = 360°/NE_r \tag{5-1}$$

式中　E_r 为转子齿数；N 为运行拍数，N = km（m 为步进电动机的绕组相数；k = 1 或 2，取决于绕组的通电方式）。

（2）种类。除了反应式步进电动机，还有永磁式和感应子式（又叫混合式）步进电动机。反应式步进电动机，除了图 5-24 所示的结构外，还有其它结构。铁心有单段式、多段式定子；磁路有径向、轴向；绕组相数有三相、四相、五相等等。

反应式步进电动机结构简单，步距角较小（可做到 1°，甚至更小），精度容易保证，起动和运行频率高，但效率较低，动态性能差，断电后无定位力矩。

永磁式步进电动机出现在 20 世纪 80 年代初。它出力大，动态性能好，但步距角大。

混合式步进电动机综合了反应式、永磁式步进电动机两者的优点，图 5-25 示的为永磁感应子式步进电动机基本结构。

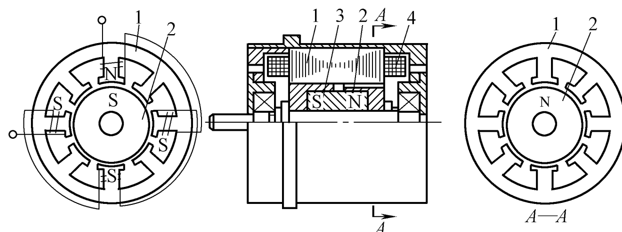


图 5-25　永磁感应子式步进电动机基本结构

1—定子　2—转子　3—永磁铁　4—绕组

它的定、转子铁心结构与反应式步进电动机相似。转子由左右两段铁心组成，两段铁心轴向错开半个齿距，中间嵌有永磁铁。这样，在永磁铁 S 端的铁心呈 S 极，永磁铁 N 端的铁心呈 N 极。当定子绕组激励时，S 极的转子铁心小齿与定子 N 极 1、5 小齿相对，而与定子 S 极 3、7 小槽相对；在转子 N 极的右段铁心则正相反。

当绕组 1、3、5、7 断电，绕组 2、4、6、8 通电时，定子 2、6 呈 N 极，4、8 呈 S 极。这时，S 极的转子铁心小齿与定子 2、6 小齿相对，而与定子 4、8 小槽相对，转子顺时针转动了 1/4 齿距的角度；同样，N 极的转子铁心段也同方向转动了 1/4 齿距的角度。

永磁感应子式步进电动机兼有永磁式步进电动机与反应式步进电动机两者的优点，步距角小，精度高，工作频率高，且功耗小，效率高。

（3）特性。定子相励磁时，步进电动机产生一个静转矩（即电磁转矩），对永磁式步进电动机，转子被推到和定子磁场一致的位置方向；对反应式或永磁感应子式步进电动机，转子转到磁路磁阻最小的位置方向。这个转矩大小按一定周期变化。图 5-26 示的为反应式步进电动机的矩角特性曲线。图中 p 为转子极对数。

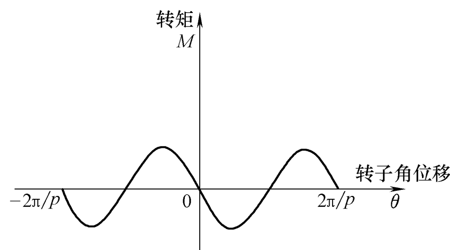


图 5-26　反应式步进电动机的矩角特性曲线

这一特性曲线表示了在单脉冲及通电电流不变情况下，步进电动机静转矩与转子角位移之间的关系。永磁式步进电动机的矩角特性基本上是正弦的，反应式步进电动机的矩角特性基本上是非正弦的。

矩角特性对步进电动机的运行十分重要。是步进电动机的基本特性。它表示若有一足够大的外转矩，使转子偏离起始点足够远，这时静转矩降低，转子将不再能回到最初的起始点，而移到新的平衡位置。

步进电动机从第一个脉冲信号的通电状态转换到第二个脉冲信号的通电状态，矩角特性曲线也由原来的特性曲线转变到新的特性曲线。随着控制频率的增加，电动机的电磁转矩将降低。在负载惯量一定时，控制频率与负载转矩间的关系称矩频特性，它包括起动矩频特性和运行矩频特性。

步进电动机矩角特性中的转矩越大，带负载转矩和加速负载转矩的能力就越大。然而，这将使转子转

动惯量增加，定子绕组电流增大。电流大，使定子热损耗增加。

步进电动机的这些特性在选用时是必须考虑的。

（4）细分。以上介绍的步距角，其实不一定是步进电动机实际工作时的真正步距角。真正的步距角还与驱动器有关，可通过驱动器对其“细分”。电动机出厂时给出了一个步距角的值，如 86BYG250A 型步进电动机给出的值为 0.9°/1.8°（表示半步工作时为 0.9°、整步工作时为 1.8°），这个步距角可以称之为步进电动机固有步距角，但对其细分后的步距角要小些。表 5-14 示的为 86BYG250A 步进电动机固有步距角与真正步距角的对应关系。

表 5-14　86BYG250A 步进电动机固有步距角与真正步距角对应关系

电动机固有步距角	所用驱动器类型及工作状态	步进电动机运行时的真正步距角
0.9°/1.8°	驱动器工作在半步状态	0.9°
0.9°/1.8°	细分驱动器工作在 5 细分状态	0.36°
0.9°/1.8°	细分驱动器工作在 10 细分状态	0.18°
0.9°/1.8°	细分驱动器工作在 20 细分状态	0.09°
0.9°/1.8°	细分驱动器工作在 40 细分状态	0.045°

从表 5-14 可以看出：步进电动机通过细分驱动器的驱动，其步距角变小了。如驱动器工作在 10 细分状态时，其步距角只为步进电动机固有步距角的 1/10。也就是说：当驱动器工作在不细分的整步状态时，控制系统每发一个步进脉冲，步进电动机转动 1.8°；而用细分驱动器工作在 10 细分状态时，步进电动机只转动了 0.18°。

细分功能完全是由驱动器靠精确控制步进电动机的相电流所产生的，与步进电动机无关。细分后的主要优点是：可消除步进电动机的低频振荡；可提高了步进电动机的输出转矩，尤其是对三相反应式步进电动机，其转矩比不细分时提高约 30% ~40%；可提高了步进电动机的分辨率，由于减小了步距角、提高了步距的均匀度。

（5）选用。主要考虑步距角、静转矩及电流三大要素。

1）步距角的选择。考虑细分后应等于或小于与每个脉冲当量对应的步进电动机转角。

2）静转矩的选择。选择的依据是步进电动机负载，而负载可分为惯性负载和摩擦负载两种。直接起动时两种负载均要考虑，加速起动时主要考虑惯性负载，恒速运行只要考虑摩擦负载。一般情况下，静转矩应为摩擦负载的 2 ~3 倍。

3）电流的选择。可根据矩频特性曲线，选择步进电动机的电流。

4）转矩与功率换算。步进电动机一般在较大范围内调速使用、其功率是变化的，一般只用转矩来衡量，转矩与功率换算如下：

$$P=\Omega\cdot M$$

$$\Omega=2\pi\cdot n/60$$

$$P=2\pi nM/60$$

式中 P 为功率（W），单位为瓦，Ω 为角速度（rad），n 为每分钟转速；M 为转矩（N·m）。

$$P=2\pi fM/400(\text{半步工作})$$

式中 f 为每秒脉冲数（简称 PPS）。

（6）注意事项。根据使用者的经验，选用与使用步进电动机有若干注意事项，现转录如下，供参考。

1）步进电动机应用于低速场合，每分钟转速不应超过 1000r，不然可通过减速装置使其在此转速下工作。因在此转速下时步进电动机工作效率高，噪声低。

2）步进电动机最好不使用整步状态，整步状态时振动大。

3）由于历史原因，只有标称为 12V 电压的步进电动机使用 12V 外，其它步进电动机的电压不是驱动电压。实际可根据驱动器选择驱动电压（建议：对 57BYG 采用直流 24 ~36V，对 86BYG 采用直流 50V，对 110BYG 采用高于直流 80V），当然 12V 的电压除 12V 恒压驱动外，也可以采用其它驱动电源，不过要考虑温升。

4）转动惯量大的负载应选择大机座号电动机。

5）步进电动机用在高速或大惯量负载时，一般不在工作速度下起动，而采用逐渐升频提速，以避免步进电动机失步、减少噪声，同时可提高定位精度。

6）高精度时，应通过机械减速，提高电动机速度。

7）步进电动机不应在振动区内工作，否则可改变电压、电流或增加阻尼。

8）步进电动机在 600 脉冲/s（0.9°）以下工作，应采用小电流、大电感、低电压驱动。

9）应先选步进电动机，后选驱动方式。

（7）除了旋转步进电动机，还有直线步进电动机。直线步进电动机有反应式和索耶式两类。索耶式直线步进电动机如图 5-27 所示。

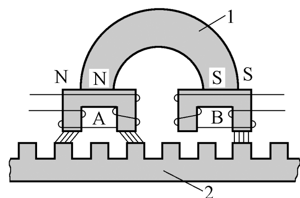


图 5-27　索耶式直线步进电动机

1—动子　2—反应板

从图 5-27 可知，它由静止部分（称为反应板）和移动部分（称动子）组成。反应板由软磁材料制成，在它上面均匀地开有齿和槽。步进电动机的动子由永久磁铁和两个带绕组的磁极 A 和 B 组成。动子是由气垫支承，以消除在移动时的机械摩擦，使步进电动机运行平稳并提高定位精度。这种步进电动机的最高移动速度可达 1.5m/s，加速度可达 2g，定位精度可达 20μm 以上。

由两台索耶式直线步进电动机相互垂直组装就构成平面电动机，如图 5-28 所示。给 x 方向和 y 方向两台电动机以不同组合的控制电流，就可以使电动机在平面内做任意几何轨迹的运动。大型自动绘图机就是把计算机和平面电动机组合在一起的新型设备。平面电动机也可用于激光剪裁系统，其控制精度和分辨率可达几十微米。

（8）内装转子位置传感器的步进电动机。传统的步进电动机是开环工作的。但为了可靠，新型步进电动机有的也内装位置传感器。有此传感器，再通过闭环控制，在其运转过程，可对转速及转量进行监控。如发生连续过载时，还会报警。即使在剧烈的负载变化或急速加速的情况下也不会失步。图 5-29 所示的就是该步进电动机工作原理框图。

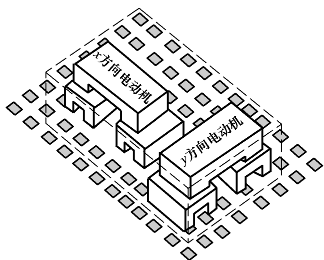


图 5-28　平面电动机

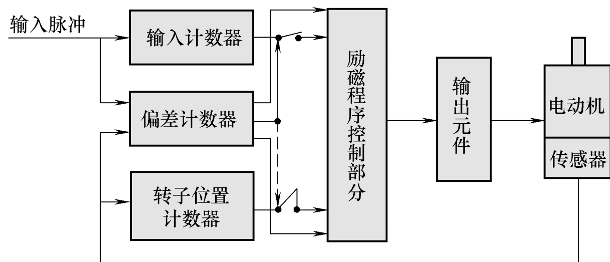


图 5-29　内装转子位置传感器的步进电动机原理框图

2. 伺服电动机

（1）概述。伺服电动机又称为执行电动机。其功能是将电信号转换成转轴的角位移或角速度，分交、直流两类，主要用于辅助运动控制。

直流伺服电动机具有良好的线性调节特性及快速的时间响应。20 世纪 70 年代以来，直流伺服电动机应用非常广泛。近年来，交流伺服技术进步很快，已用得越来越多，但在有些方面还是直流伺服电动机较适用，比如：需要高度平滑的运转，特别是在低速时；需要高速度（>5000r/min）；需要特别的速度稳定性；较恒定的力矩；需要直流电源输入的场合。当然它在缺点方面也很明显，如有刷伺服电动机要维护更换电刷（无刷伺服电动机目前功率一般较小，性能不如有刷伺服电动机）；一般不用于高精度位置控制，用于速度和力矩控制较合适。

交流伺服电动机工作原理与交流异步电动机相同。定子上有两个相空间位移 90°电角度的励磁绕组 W_f 和控制绕组 W_c。励磁绕组 W_f 接一恒定交流电压，利用施加到控制绕组 W_c 上的交流电压或相位的变化，

达到控制伺服电动机运行的目的。

交流伺服电动机常用的转子结构有笼型和非磁性杯形。笼型转子与异步电动机的笼型转子结构相似，但较细长，其励磁电流小，功耗低，体积小，机械强度高，广泛应用于交流控制系统。非磁性杯形转子是用非磁性金属如铝、紫铜等制成。这种转子惯量小，运行平稳，噪声小，灵敏度高，主要用于一些要求运行平滑的系统。交流伺服电动机分为同步型和异步型 AC 伺服电动机两种。永磁转子的同步伺服电动机由于永磁材料不断提高，价格不断下降，控制又比异步伺服电动机简单，容易实现高性能的缘故，所以永磁同步电动机的 AC 伺服系统应用更为广泛。

伺服电动机实际上可组成一个闭环运动控制系统，除了电动机，还有驱动器。驱动器是电力电子电路。接收脉冲或电压信号，生成驱动伺服电动机的工作电源，使伺服电动机转动，同时接收来自伺服电动机的反馈信号。

伺服电动机除了动力部分，接受驱动器电源作用而转动，同时还有反馈器件，可间接把位置、速度、转矩等信号回馈给驱动器。

目前常用的位置和速度检测反馈器件有光电式和电磁式两种。例如光电编码器、磁编码器、旋转变压器（BR）以及多转式绝对值编码器。后面两种，可作多种检测功能应用，既可检测系统位置和转子转速，又可检测转子磁极位置。它坚固耐用，不怕振动，耐高温，惟存在信号处理电路复杂缺点。

正因为伺服电动机实际可组成一个闭环运动控制系统，所以功能、性能都比较高，可用于速度控制、转矩控制、位置控制及其组合。

（2）特点。可组成交流伺服电动机的特点有：

1）运行稳定。交流伺服电动机与一般异步电动机相比，具有更大的转子电阻，它的机械特性的斜率都是负值。转速随转矩的增加而均匀下降，因此可在 n 为 0 ~ n0（空载转速）之间稳定运行。这样的机械特性决定了伺服电动机的效率较低。

2）可控性好。两相交流伺服电动机机械特性的斜率为负值，在单相供电时，转矩和转速符号相反，因此，当单相励磁时，伺服电动机不会发生自转现象，即控制信号一旦消失，电动机立即停转。

3）快速响应。控制绕组接到控制信号后，能快速起动，信号消失后，又立即自行制动、停转。一般用机械的和电的两个时间常数来表征，时间常数小，快速响应良好。一般要求电动机具有高堵转转矩、小的转子惯量及电感和电阻的比值。

4）灵敏度高。伺服电动机具有小的起动电压。起动电压指的是在额定励磁电压下，加于控制绕组以使电动机开始连续转动的最小电压。起动电压愈低，则灵敏度愈高，系统的不灵敏区愈小。一般起动电压应小于额定电压的 3% ~4%。

5）机械特性和调节特性的非线性度指标严格。机械特性的非线性度 K_m 是指在额定励磁电压下，对任意控制电压时的实际机械特性与线性机械特性在转矩 M 等于 $M_k/2$ 时的转速差 Δn（图 5-30 所示为交流伺服电动机机械特性的非线性度）与空载转速 n_0 之比的百分数，即 $K_m = \Delta n/n_0 \times 100\%$。$K_m$ 要求小于 10% ~15%。

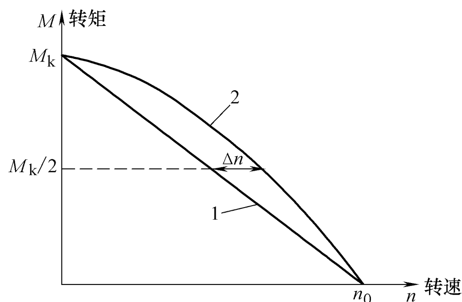


图 5-30　交流伺服电动机机械特性的非线性度
1—实际机械特性　2—线性机械特性

机械特性的非线性度 K_r 是指在额定的励磁电压和恒定的转矩下，当实际控制电压的百分值 a_e 等于 0.7 时，实际机械特性与线性机械特性转速差 Δn 与空载转速 n_0 之比的百分数，即 $K_r = \Delta n/n_0 \times 100\%$。$K_r$ 要求小于 15% ~25%。

直流伺服电动机，其工作原理与一般直流电动机相同，其转速 n 为

$$n = E/(K_1 \Phi) = (U_a - I_a R_a)/(K_1 \Phi)$$

式中，K_1 为常数。改变电枢电压 U_a 或改变磁通 Φ，均可控制直流伺服电动机的转速，但一般采用控制电

枢电压的方法。在永磁式直流伺服电动机中，励磁绕组被永久磁铁所取代，磁通 Φ 恒定。

（3）选用。选用步进电动机，还是选用伺服电动机，是首先要考虑的。应根据具体应用情况而定，因各有其特点，可参考表 5-15 进行选择。

如果选用伺服电动机，应注意如下问题：

1）伺服电动机的最高转速应严格控制在电动机的额定转速之内。

2）为了保证足够的角加速度使系统响应灵敏和满足系统的稳定性要求，负载转动惯量 J_L 应限制在 2.5 倍伺服电动机转动惯量 J_M 之内。

表 5-15　步进电动机与伺服电动机比较

	步进电动机	伺服电动机
转矩范围	中小转矩(一般在 20N·m 以下)	小中大,全范围
速度范围	低(一般在 2000r/min)以下,大转矩电动机小于 1000r/min	高(可达 5000r/min),直流伺服电动机更可达 1~2 万 r/min
控制方式	主要是位置控制	位置/转速/转矩方式一体
平滑性	低速时有振动(但用细分型驱动器则可明显改善)	好,运行平滑
精度	一般较低,细分型驱动时较高	高
矩频特性	高速时,转矩下降快	转矩特性好,特性较硬
过载特性	过载时会失步	可 3 倍以上过载(短时)
反馈方式	一般为开环控制,也可接编码器	闭环方式,编码器反馈
编码器类型		光电型旋转编码器(增量型/绝对型),旋转变压器型
响应速度	一般	快
耐振动	好	一般(因为光电编码器较怕振动)
耐温度	好	一般
维护性	基本可以免维护	较好
价格	低	高

3）空载加速转矩一般应限定在变频驱动系统最大输出转矩的 80% 以内。

4）在正常工作状态下，负载转矩不应超过伺服电动机额定转矩的 80%。

5）连续过载时间应限制在伺服电动机规定过载时间之内。

5.3　闭环控制

关键词：内置高速计数输入点、高速计数单元、表比较、范围比较、高速计数器复位（硬件复位、软件复位）、脉冲量输入模拟量输出闭环控制、模糊量输入脉冲量输出闭环控制、脉冲量输入脉冲量输出闭环控制、开关量输入其它量输出闭环控制

这里讲的闭环控制主要是使用了脉冲量。为此，它的程序应在输入、输出的环节上，要作适当处理。以下分脉冲量入开关量输出、脉冲量输入模拟量输出、模拟量输入脉冲量输出、脉冲量输入脉冲量输出及开关量输入其它量输出 5 种闭环控制进行讨论。

此外，如果使用绝对式编码器检测运动部件的位置，其示值是用开关量输入点读入。所以，它的入是开关量。控制输出可以是开关量、模拟量或脉冲量。用其不同组合可实现闭环位置控制。本节也将对其作简要介绍。

5.3.1　脉冲量输入开关量输出闭环控制

脉冲量输入开关量输出闭环控制使用的是高速计数比较控制。它采集脉冲累计数，并设

定值进行比较。根据比较结果，产生控制输出。其特点是，全过程都是或都可以中断方式工作。信号采集及比较中断工作，控制输出中断产生。

高速计数比较控制多用于行程控制，以控制运动部件的行程，达到如控制下料长度、控制部件位置等要求。

1. 小型 PLC 高速计数闭环控制

用小型机进行高速计数比较控制，首先要使它能高速计数。为此，要作好 3 个设定：

（1）是否使用高数计数功能？应设为使用。

（2）复位方式，是软件复位，还是 Z 相加软件复位（如 CQM1 机，用特殊继电器 252. 00 ON，使高速计数器现值回到 0）？如需硬件复位，选后者。

（3）高速计数模式，是增计数？还是两相（加/减）计数？等等，按自己要求选定。

这些选定都是改变 DM 参数区（DM6642）的值实现，也可在 CXP 软件的设定窗口上选择，而后者实质也是改变 DM 参数区的值。

有了以上设定，PLC 运行时，即可脉冲从高速计数输入点读入脉冲。只要向 PLC 送入脉冲，高速计数器的现值即有变化。高速计数器为两个字，248、249 的内容即为它的现值，也可用 PRV 或 INI 指令读、写它的现值，有关细节，可参阅有关说明书。

读入脉冲后，怎么处理？比较，根据比较的结果实施控制。有表比较及范围比较。为此，要起动相应指令。图 5-31 所示为使用范围比较及起动比较的程序。这里用 CQM1 的高速计数功能，显示及控制某金矿提升机的位置，以代替原用的行程开关控制。该系统是原黄金学院自控系开发的，已收到了良好的效果。

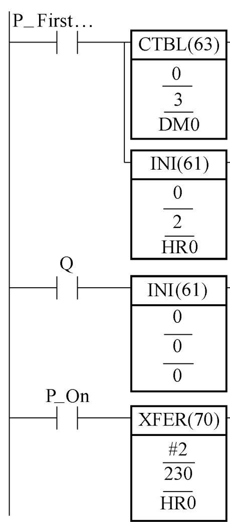


图 5-31　范围比较及起动比较

图 5-31 中用 P_First…为特殊继电器，在程序扫描第一周期 ON，常用于程序初始化。在此，为调 CTBL 指令及 INI 指令。CTBL 用以建立范围比较，被比较值首字存于 DM0 中。执行 INI 指令，是把 HR0 通道的内容传给高速计数器，置它的现值。

CTBL 指令有三个操作数。第一个默认为 0，第二个为控制字，分别取值为 000、001、002、003，第三个为字地址（TB），存储被比较数。

C 的含义为：

000：建立表比较，并开始比较

001：建立范围比较，并开始比较

002：建立表比较，由执行 INI 指令起动比较

003：建立范围比较，由执行 INI 指令起动比较

字地址的含义：

若为表比较：可对 16 个双字比较，这里 TB 及随后的字的含义为：

TB：指明与多少个字比较，取值为 1 ~ 16

TB + 1：目标值低 4 位

TB + 2：目标值高 4 位

TB + 3：当现值与目标值相等时将调用的子程序号

这相邻的 3 个字算一组，接着还可设第二组，最多可设 16 组，占 48 个字。加上 TB，

最多时，从 TB 开始到 TB +48 的字都要用。

若为范围比较，则固定用 8 个范围，其含义为：

TB：低限，低 4 位

TB +1：低限，高 4 位

TB +2：高限，低 4 位

TB +3：高限，高 4 位

TB +4：当现值落入上述范围，将调用子程序号，可为 000 ~ 127

这里每组用 5 个字，必须设 8 组，共用 40 个字。如设了，但又不用，则应把调子程序号那个字设为 FFFF。

从对 CTBL 指令的介绍可知，本例用的是仅建立范围比较，而是否执行比较？由执行 INI 指令起动。

INI 为中断指令，也有 3 个操作数。

第一个，默认为 0

第二个，可为 0、1、2、3。0 为起动比较，1 为停止比较，2 为现值更新，3 为脉冲输出停止（不用于此场合）。

第三个，默认为 0，但当第二个为 2 时，它及它的高一个字存储的是向高速计数器更新值。

如图 5-31 所示程序，高速计数值与比较值相等所调的子程序号，靠 DM0004、DM0009、…确定。而比较值，靠 DM0000、0001，DM0002、0003，…确定。

本例 INI 执行取决于“Q”是否 ON，ON，意味系统工作，即执行 INI 指令，起动范围比较。

利用 CQM1 高速计数功能，目的为控制矿车运动。当矿车接近层面时要减速，并显示层的数据，便于工人操作，使其准确停下来。

这里的对子程序内容及 DM0000 ~ DM0039 的设定细节，就不多介绍了。

另外，这里还时时把 SR230、SR231（它是特殊继电器存的为高速计数器的内容，CPM 机为 248、249）的内容传给 HR00、HR01，目的是使 HR00、HR01 始终保存着计数器现值。HR 有掉电保持功能。上电时，再把它的内容传至高速计数器（不然，高速计数器总是从零开始计数），可反映上次停车时矿车的实际位置，很便于矿车操作。

提示：本例用的为 CQM1机，其它机型也有类似功能，也可使用，但具体内部器件及输入、输出地址及指令的细节可能有所区别。使用时可参考有关手册操作。

2. 高速计数模块闭环控制

用高速计数模块进行高速计数，作比较控制，为此要：

（1）高速计数模块类型较多，要选合适的类型。

（2）计数模式也很多，也要按需要选用。

（3）进行硬件设定。设定机号，不能与其它特殊单元重复；设定 DIP 开关，与模式、输入脉冲、复位、控制信号的使用等情况对应。

（4）正确安装及接线。

（5）进行参数设定。对计数模式的设定应与硬件的设定一致，否则不能工作，并显示出错信号。不同的计数模式，或相同的计数模式而情况不同，参数的设定都不可能完全相同。

（6）选用命令及使用好标志，正确编制梯形图程序，使高速计数模块实现所要求的功能。

例：图 5-32 所示为使用高速计数模块的例子，用以控制剪接材料的长度。

图 5-32 中 1 为导轮，由电动机带动，旋转时，可使材料从卷料中绕出，并送向切刀 7。2 为旋转编码器，随着细杆送进，而发出计数脉冲，经信号线送高速计数模块。高速计数模块依计数情况，可产生#0、#1、#2 及#4 输出，以分别控制制动松开和电动机工作、低速到高速转换、高速到低速转换及制动和切刀切下等动作。

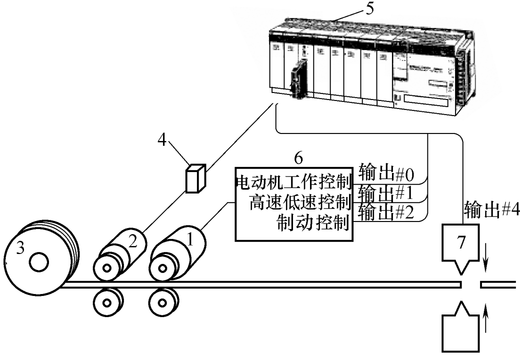


图 5-32　高速计数模块使用实例

1—导轮　2—编码器　3—卷料　4—编码器用接线器　5—高速计数模块　6—电动机控制器　7—切刀

本例的硬件设定为：机号为 1，模式为 3，背板上 DIP 开关针 3、5 ON，其它 OFF。设成内部复位有效及输入脉冲乘 2。

软件设定：主要对 DM，因机号为 1，故占用的 DM 是 DM1100 ~ DM1199。

图 5-33 所示为工作过程的信号波形。表 5-16 所示为该图中有关参数选定。所有参数的单位都是 ms。如 A100ms，其对应的计数值应是 100 乘脉冲频率，再乘 1000。

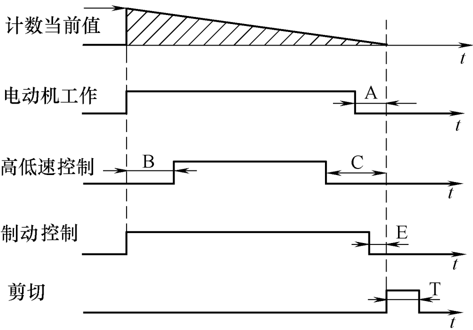


图 5-33　工作过程信号波形

表 5-16　参数值

参　　数	ms
预置计数值	10000
A	100
B	150
C	200
D	0
E	50
T	500

参数值的选定与准备剪切材料的长度、工艺参数有关，这也是程序调试过程要解决的问题。

它的四个输出为：#0 ON 对应于制动松开和电动机起动；#1ON 时为快速；#2ON 使制动到给定位置，即计数值为零时，#4 输出 ON，使切刀剪切。

DM 区设定见表 5-17。

表 5-17　DM 字设置

DM 编号	各数位取值				备注	DM 编号	各数位取值				备注
DM1100	0	3	0	0	设模式 3	DM1104	0	0	0	0	
DM1101	0	1	0	0	A = 100ms	DM1105	0	2	0	0	C = 200ms
DM1102	0	0	0	0		DM1106	0	0	0	0	
DM1103	0	1	5	0	B = 150ms	DM1107	0	0	0	0	D = 0

（续）

DM 编号	各数位取值				备注	DM 编号	各数位取值				备注
DM1108	0	0	0	0		DM1112	—	—	—	—	
DM1109	0	0	5	0	E = 50ms	DM1113	—	—	—	—	
DM1110	0	0	0	0		DM1114	0	0	0	0	预置 = 10000ms
DM1111	0	0	5	0	T = 500ms	DM1115	1	0	0	1	仅输出#4

与上述对应的梯形图程序如图 5-34 所示。

从图 5-34 可知，这个梯形图程序并不复杂。按下“复位按钮”，110.06 ON，使计数器及输出复位。按下“起动按钮”，110.00 ON，使计数起动。“常通标志”，110.02 ON，使输出使能，等等。输出 0、1、2、4 是高速计数单元的输出点的输出，PLC 无需，也不可能用程序去控制。

为了检测工作后什么时候完成，用微分下降指令检测输出 4（控制切刀动作的）从 ON 到 OFF（表明整个工作完成），其操作数为 30.00。工作完成，它将 ON 一个扫描周期。这足以使 30.01 ON，并自保持。这表明操作完成。重新起动时，110.00 的常闭触点断，使 30.01 复位。

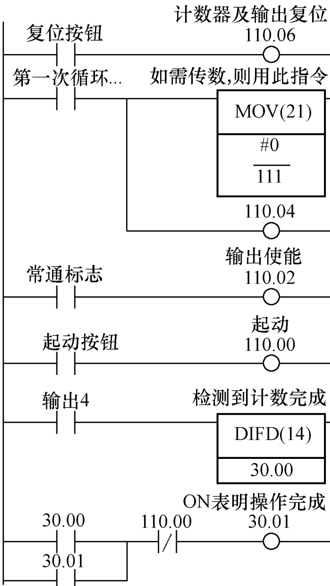


图 5-34 起动结束梯形图程序

从上例可知，高速计数模块比较控制程序设计的工作量不大、主要工作是硬件、软件设定及参数选择。很多经验证明，首先要设定好，不出现报警，确保脉冲数读入；其次是参数选定，能进行比较及产生控制输出。

提示：本例用的为 C200H 机的高速计数模块。其它机型的模块很多，也都可使用，但模块的功能、特性、内部器件及输入、输出地址及使用细节可能有所区别，使用时可参考有关手册操作。

5.3.2 脉冲量输入模拟量输出闭环控制

脉冲量输入模拟量输出的闭环控制，可以是 PID 控制，也可是模糊控制，可依情况选定。在选用某种控制时，控制参数也可作不同的选定。

在此要补充说明的是，在相同的系统中，根据不同的偏差值，选用不同的控制算法及不同的控制参数，以提高控制效果，是完全可能的。如在偏差很大时，根据经验，干脆输出很大的控制量；而到了调节量接近给定值时，转换用 PID 控制，慢慢使其达到控制要求。

只是，在算法切换时，或参数变换时，其控制输出不应突变，以避免对系统的冲击。为此，在使用多算法、多参数控制时，各种算法应相互追踪，一旦切换或转换，控制输出不产生突变。

以下以直流电动机转速控制为例，看怎样使用 PID 进行闭环控制。系统的原理框图如图 5-35 所示。

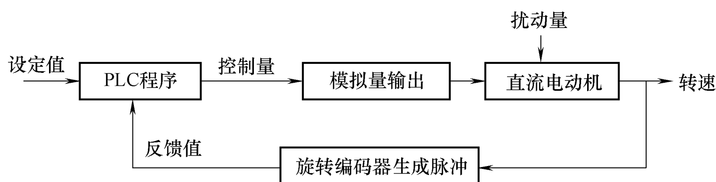


图 5-35　PI/AO 闭环控制原理框图

从图 5-35 可知，这里一个重要问题是，如何读入旋转编码器产生的脉冲信号，并将其转换为频率信号，以用作转速反馈值。而有了这个反馈值，加上给定值，再采用某个控制算法计算，即可得出控制量。如果控制算法合适，选用的控制参数得当，则可有效地实现系统闭环控制。

图 5-36 所示为脉冲读入及频率计算程序，该程序运行于 CPM2A 机，设备用的是沈阳旭风电子科技开发有限公司的 SAC-PLC-TS4 测试台。

图 5-36 中，STIM 为定时中断设定及启用指令。它用“P_First_C”（仅在第一个扫描周期 ON）作为执行条件。两指令配合使用，使定时中断初始化并启用。选定这些操作数的 STIM 指令含义是，每隔 100ms 定时中断一次，中断时调子程序 1。

从图 5-36 中子程序 1 知，它用 XFER 指令，把高速计数器的现值（CPM2A 机高速计数器现值存于通道 248、249 中）传送给“BCD 脉冲频率”。接着，使 252.00 ON，使高速计数器复位（高速计数器预设成仅用软件复位）。再接着，把“BCD 脉冲频率”转换成“二进制脉冲频率”。当然，为使 CPM2A 机具有高速计数功能，还得作相应设定。

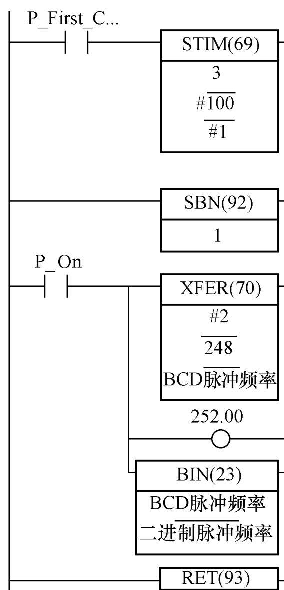


图 5-36　脉冲读入及频率变换

可知，这里“BCD 脉冲频率”为每 100ms 采集的脉冲数。因为 CPM2A 机的高速计数器用 BCD 码表达，故 XFER 指令执行后得到的就时 BCD 码表达的脉冲频率。

由于要使用 OMRON 公司的 PID 指令实现闭环控制，而 PID 指令用的反馈值须用二进制码表示，故这里紧接着又作了 BCD 码到二进制码的转换。

图 5-37 所示为使用 PID 指令实现闭环控制的梯形图程序。

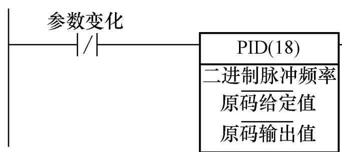


图 5-37　PID 闭环控制程序

从图 5-37 知，该程序很简单。仅一个执行条件，一条执行指令。这里“参数变化”何意？当重新选定参数时，它 OFF 一个周期，以使所选定的新参数生效。这是因为 OMRON C 系列机 PID 指令执行后，改变参数无效，用这么处理就能生效了。

执行 PID 指令后的“原码输出值”也是二进制值，故不必转换，直接就用模拟量输出通道就可以了。

要补充的是，最好能依据误差变化，选择不同的控制参数，以提高控制效率。图 5-38 所示为此类程序。

从图 5-38 可知，偏差值大于“很大偏差”时，LR1.10 ON，并使 LR1.01 置位；偏差值小于“很大偏差”时，LR1.10 OFF，并使 LR1.02 置位。这两种情况赋给 PID 指令不同的控制参数。该程序用了 DIST 指令。它是偏移传送指令，执行时，把第 1 个操作数传送给第 2 个操作数地址加第 3 个操作数形成的 DM 地址。如图 5-38 所示，如偏差大于“很大偏差”时，将把 500 赋给“原码给定值”+1的 DM 地址，即存储“比例带”参数的地址；而偏差值小于“很大偏差”时，则把 2000 赋给它，使控制作用弱些。还可用其它第 3 个操作数的不同 DIST 指令，以进行其它参数修改。只是该图未把要改的参数全部列出。

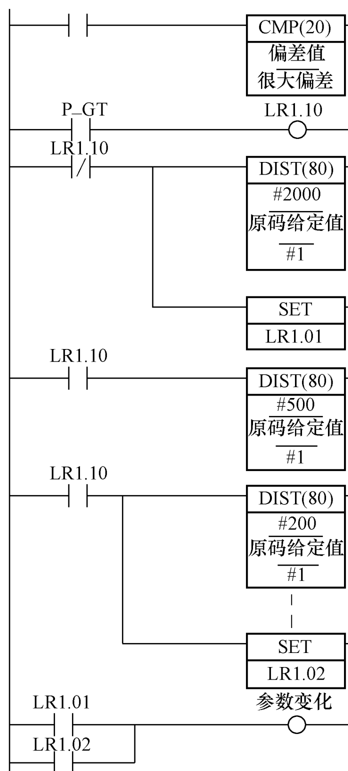


图 5-38　参数改变

从图 5-38 还可知，从“大于”变“小于”，或从“小于”变“大于”过程，“参数变化”将 ON，进而其常闭触点使 PID 指令停止执行。为使停止执行后，接着还能继续工作，故须把图 5-39 所示的小程序置于执行 PID 指令之后。即 LR1.01 、LR1.02 只能同时 ON 一个周期，即“参数变化”只能 ON 一个周期。而有了这一个周期的 ON，已足以使新参数生效了。

如果有更多的 PID 参数选定，也可按此思路进行设计。

有的 PLC 及 OMRON CS 系列机的 PID 指令执行后，参数可即改，即其作用就不需进行上述处理。

提示：本例用的为 CP2M 机，所控制是速度，其它机型也有类似功能，也可使用。

5.3.3　模拟量输入脉冲量输出闭环控制

这种闭环控制反馈输入的是模拟量，而控制输出是脉冲量。脉冲量可以是不同的输出脉冲数，不同的脉冲频率或不同的脉宽。图 5-40 所示为模拟量输入脉冲量输出的电阻加热炉温度闭环控制框图。

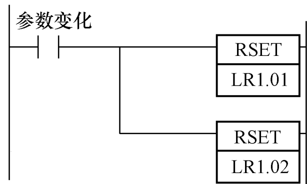


图 5-39　复位小程序

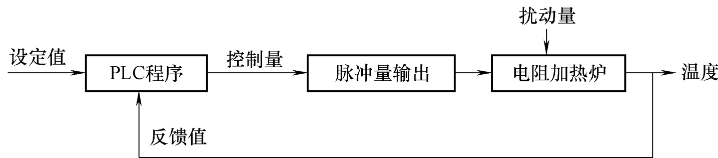


图 5-40　电阻加热炉温度闭环控制框图

从图 5-40 可知，它的输入与模拟量控制时的输入相同，输出要用到脉冲量。现以使用脉宽调制的脉冲输出为例，介绍它的 PID 控制有关程序。

图 5-41 所示为模拟量输入初始化程序。该程序用于 CPM2A 机及 CPM1A_MA002 模拟量模块。设备用的是沈阳旭风电子科技开发有限公司的 SAC-PLC-TS3A 测试台。

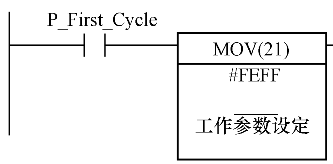


图 5-41　模拟量输入初始化

从图 5-41 可知，该初始化的目的把常数 FEFF 赋值给“工作参数设定”通道。参阅 CPM1A_MA002 说明书知，它使各模拟量输入通道可用电压或电流输入，范围为 1 ~5V 或 1 ~1.5mA，而且输入值已作了求平均滤波。

图 5-42 所示为控制及输出转换程序。由于控制电路的原因，与 5-3-1 节所示的例子不同的是，它的参数设定设为正反馈控制，即反馈值大，控制输出也大。这在程序上无法体现，只是在此说明。PID 的参数也可用 5-3-1 节所示的例子处理，在此不再重复说明，故该图只是整个程序的控制及输出部分。

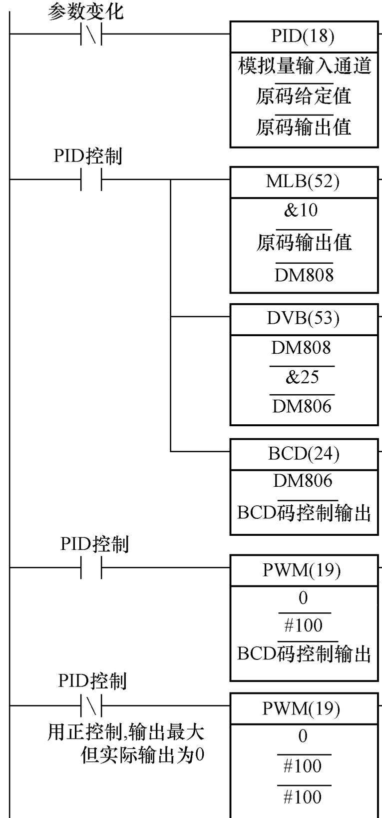


图 5-42　控制及输出转换

从图 5-42 可知，控制程序也仅两条指令，PID 及其执行条件。PID 指令的输入、输出均为二进制码。反馈输入，即 PID 指令的第 1 操作数，直接用模入通道。输出，即它的第 3 操作数，也是二进制码，在 0 到 FF 间变化。而脉宽调制 PWM 指令控制输出（第 3 个操作数，它的第 1 操作数为指定发送口，第 2 操作数为指定脉冲频率的 10 倍）用的是 BCD 码，而且只能在 0 到 100 间变化。故从 PID 计算得到输出到 PWM 的真正输出需进行转换。

该程序的 MLB（二进制运算乘 10），DVB（二进制运算除 25），及 BCD 指令就是用于实现这个转换。

最后不进行 PID 控制时，PWM 输出为 100，为最大值，目的是停止电阻加热炉加温。

5.3.4　脉冲量输入脉冲量输出闭环控制

这种闭环控制反馈输入的是脉冲量，而控制输出也是脉冲量。脉冲量可以是不同的输出脉冲数，不同的脉冲频率或不同的脉宽。图 5-43 所示为脉冲量输入脉冲量输出的转速闭环控制框图。

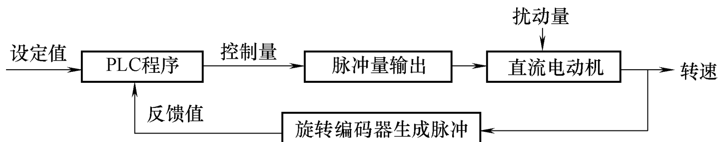


图 5-43　脉冲量输入脉冲量输出的转速闭环控制

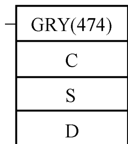
从图知，它的输入与图 5-34 所示的例子相同。如输出用脉宽调制，则与图 5-39 所示的例子相同。实际上本例输出用的正是脉宽调制。用的仍为 CPM2A 机，设备也是用的是沈阳旭风电子科技开发有限公司的 SAC-PLC-TS4 测试台。

显然，整个程序可用图 5-34 所示的例子及图 5-39 所示的例子程序有关部分的组合。两者合成，即可实现如图 5-43 所示的闭环转速控制。

5.3.5　开关量输入其它量输出闭环控制

这种组合可用于绝对式编码器输入的位置控制。当运动部件运动时，编码器的示值将发生变化。可根据示值变化的情况产生不同的控制输出，以实现部件的位置控制。所以，这里的开关量不是一个而是一组。如果为 8 位编码器，就是 8 个。

绝对式旋转编码器一般用格雷码，前已提及，格雷码是无权码，无法进行大小比较。所以使用时要先译码。CP1H 机有它的译码指令。其梯形图格式为



C——控制首字；

S——源字；

D——结果首字。

图 5-44 所示为这些字的含义图解。

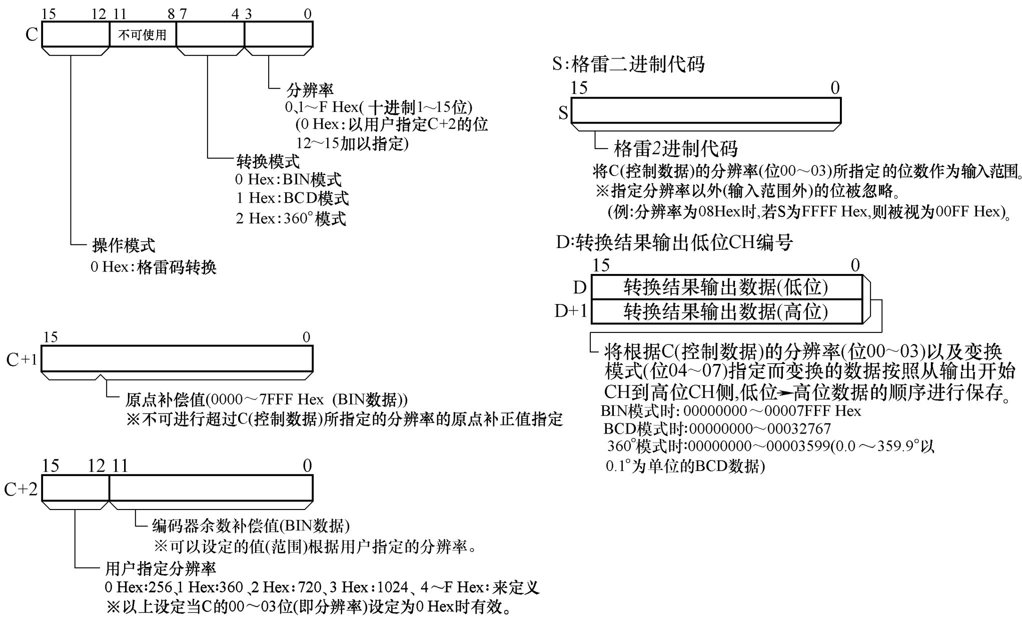


图 5-44　GRY 指令操作数含义图解

图 5-45 所示为使用 GRY 指令一例。这里，设定 8 位分辨率，转换为二进制码，原点补正为 1AHex，格雷码为 29Hex 时的译码结果。其值为 17Hex。这里的补正值指译码后减去它，所得的差为实际使用值。如本例，译码后的值为 31Hex。与补正值相减，差值为 17Hex。

开关量输入其它量输出闭环控制，与其它“输入”的控制不同的只是在“输入”上。如果从绝对式编码器输入，经使用这里介绍的 GRY 指令译码就可得到二进制码、BCD 码或角度值。这些值都可进行大小比较。进而可产生相应控制输出。而控制输出可使用开关量、脉冲量或模拟量。具体程序已有介绍，在此不再赘述。

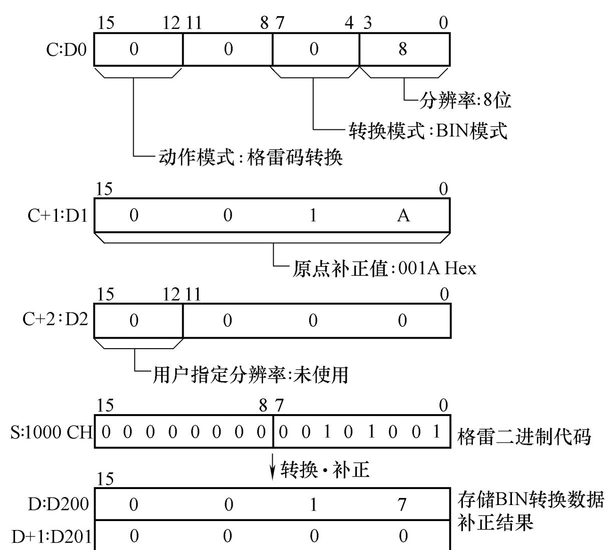


图 5-45　使用 GRY 指令例

5.4　开环控制

关键词：独立运动控制、协调运动控制、累加直线插补、逐点比较圆弧插补、直接目标追踪控制、追踪运动控制、三轴协调控制、列表数据运动控制

部件运动最多有 6 个自由度，一般用 3 个移动坐标，3 个转动坐标定位。对应的有单轴，双轴及多轴运动控制。开环控制的关键是控制好脉冲输出。

单轴运动控制是指，在各坐标上的运动互不相关，彼此独立。对这样运动系统，不管有多少坐标，从控制的角度看，都是使运动部件在一个坐标上，按要求的速度、加速运动，从一个位置移动到另一个位置。

双轴及多轴运动协调控制是指，控制对象在多轴上的位移要协调，以使控制对象能按预定的轨迹运动。协调运动控制有 3 个方法：

（1）直线圆弧插补。直线圆弧插补法常见的是把要求的运动轨迹分解为若干小段的直线或圆弧。而每小段直线或圆弧则运用一定算法一步步实现。常用的算法有逐点比较法、累加法等。

（2）目标追踪法。如果运动轨迹可用数学式子表达，也可用目标追踪控制。办法是划分好足够小的步，一步步计算位移目标，然后控制脉冲输出去追踪目标。这里“足够小的步”是指保证仅一个脉冲输出即可追上目标。

（3）列表数据控制。先由计算机计算控制对象各坐标的位置数据，下载给 PLC DM 区，PLC 根据 DM 区存储的数据控制各坐标运动。由于 PLC DM 区的增大，当位移范围不大时，这个方法是很简便的。

本章讨论的开环控制用的都是小型机。因它内置有处理脉冲量的 I/O 点及相应指令。而中、大型机，无这样的内置点与处理指令，运动控制则用特殊单元实现。这将在本书第 5.6 节讨论。

使用小型 PLC 进行运动开环控制，一定要选用具有晶体管输出的型号。因为运动控制脉冲输出的频率高，用继电器输出是不可能的。

以下将就单轴独立运动控制、双轴协调运动控制及双轴运动追踪控制分别进行讨论。

5.4.1 单轴运动控制

单轴运动控制也称点位控制。在 PCB 钻床、SMT、晶片自动输送、IC 插装机、引线焊接机、纸板运送机驱动、包装系统、码垛机、激光内雕机、激光划片机、坐标检验、激光测量与逆向工程、键盘测试、来料检验、显微仪、定位控制、PCB 测试、焊点超生扫描检测、自动织袋机、地毯编织机、晶片切割机等中都有它的应用。

1. 单段位置控制

在单轴（某个坐标）上，当工作命令发出后，可使部件按指定速度（指定脉冲频率）完成指定位移量（指定脉冲数）的位移，即这里称的“位置控制”。组成这样系统的硬件可以是：小型 PLC、步进电动机及配套设施与运动部件。其中 PLC 是通过运行程序，用输出脉冲设施这个控制。

图 5-46 所示为实现这个控制的程序。其作用是当“工作”ON 后，将使脉冲输出口发出 1000 个，频率为 10Hz 的脉冲。进而使运动部件产生相应于 1000 个脉冲当量的运动。程序的算法是：先传送控制数据进行，然后执行相关脉冲输出指令。

从图知，它先把 1000 个脉冲数传送给 DM101（高位）、DM100（低位）。然后微分执行“PULS”指令，选择 010。00 为脉冲发送口，选择独立工作模式，选择用 DM101（高位、0）、DM100（低位、1000）确定脉冲数。再把 10 传送给 DM102，并执行“SPED”指令，选择 010. 00 为脉冲发送口，选择独立工作模式，选择用 DM102 确定脉冲频率。显然，根据这组指令，即可使 PLC 向 10. 00 口发出频率为 10 的 1000 个脉冲，进而使部件按上述要求运动。

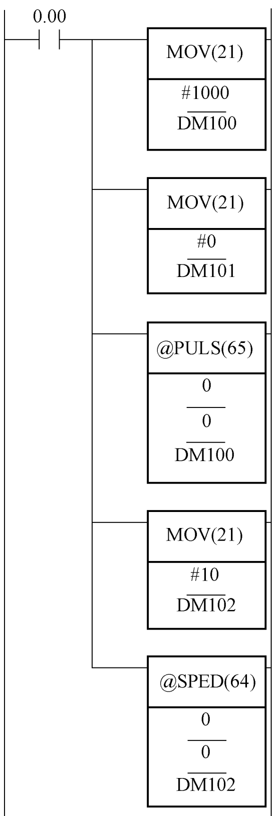


图 5-46 位置控制程序

2. 加、减速度及位置控制

在单轴（某个坐标）上，当工作命令发出后，为了工作平稳，可使部件按指定的加速度（指定脉冲频率增加率或指定加速时间）、指定的目标速度（指定输出脉冲频率）、指定减速度（指定脉冲频减小率或指定减速时间）完成指定位移量（指定脉冲数）的位移，即这里称的“加、减速度运动控制”。组成这样系统的硬件也可是：小型 PLC、步进电动机及配套设施与运动部件。PLC 则是通过运行程序，用脉冲输出口控制这个设施。

图 5-47 所示为加、减速度位置控制的一个例子。它要求用 5s 时间，把输出脉冲频率增加到 100Hz，减速时则是用 5s 时间，从 100Hz 减速到最小频率。

图 5-48 所示为三种 PLC 实现这个控制的程序。其作用是当“工作”ON 后，将使脉冲输出频率逐渐增加，5s 后达 100Hz。输出脉冲总数 20000 个。当发送脉冲接近时，减速，于 5s 后减速到最小值，并停止发送。程序的算法也是：先传送控制数据进行设定，然后执行相关脉冲输出指令。

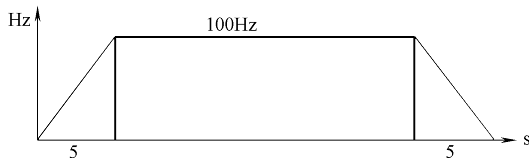


图 5-47　输出频率变化简图

图 5-48 所示为 CPM2A 用的程序。它先把 20 传送给 DM101（高位）、0 传给 DM100（低位）。然后微分执行“PULS”指令，选择 010.00 为脉冲发送口，选择独立工作模式，选择用 DM101、DM100 确定脉冲数（即指定 20000 个脉冲）。再把 20 传送给 DM110，把 100 传送给 DM111，把 20 传送给 DM112，并执行“ACC”指令，选择 010.00 为脉冲发送口，选择独立工作模式，选择用 DM110 确定脉冲频率增加率（即指定脉冲频率增加率为 20），选择用 DM111 确定脉冲频率（即指定脉冲频率为 100），选择用 DM112 确定脉冲频率减小率（即指定脉冲频率减小率为 20）。显然，根据这组指令，即可使 PLC 向 10.00 口发出脉冲，进而使部件按上述要求运动。

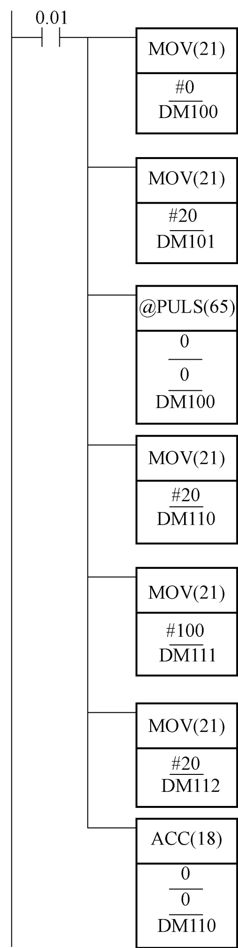


图 5-48　加、减速度位置控制程序

3. 多段位置控制

以上介绍的运动控制程序只能做一次位移。如果要多次位移怎么办？它的算法是：先设定第 1 段数据，并起动第 1 段位移，待第 1 段完成后，开始第 2 段设定，再起动第 2 段位移…直到所有位移结束。

图 5-49 所示为 2 段位置控制简图。起动后先以频率 20Hz 发送 20000 个脉冲，之后，停留 2 秒，再以频率 10Hz 发送 10000 个脉冲。

实现这个控制算法的要点是：传送脉冲数及要求频率数据；使用脉冲输出指令；判断第 1 段是否到位，如到位，再传送脉冲数及要求频率数据；再使用脉冲输出指令，直到控制完成。图 5-50 所示为实现此算法，与图 5-49 对应的 PLC 程序。

图 5-50 中：

① 起动逻辑。

② 把脉冲数 20000 的 20 传送给 DM5（高位）、DM10（低位）。

③ 把脉冲频率 20 传送给 DM6（单位为 10Hz，故传送 2）。

④ 进行第 1 段输出脉冲与要求脉冲数比较，判断是否到位。由于 CPM2A 没有脉冲输出完成的标志，这里用硬件把输出脉冲信号输入给高速计数器（000.00 点设为进行高速计数）。而特殊继电器 248、249 则记录着采集的脉冲数。故这里把 248、249 与设定脉冲数进行比较。

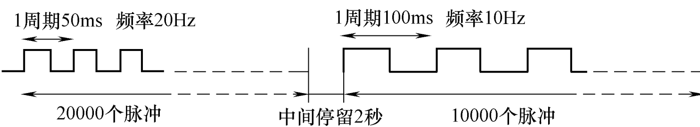


图 5-49 2 段位置控制简图

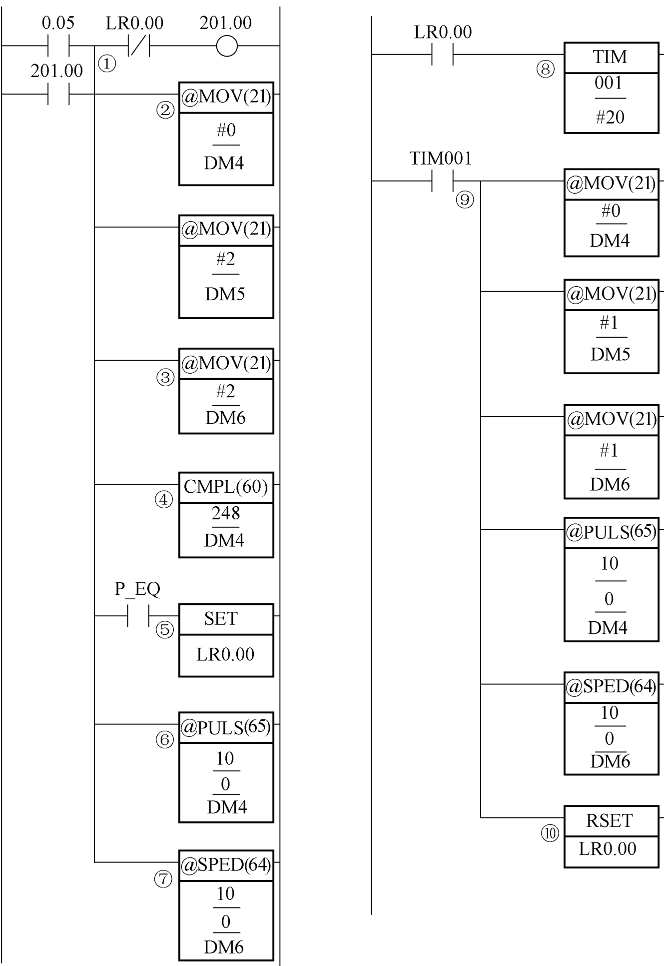


图 5-50 2 段位置控制 PLC 程序

⑤ 如比较相等，则 LR0. 00 置位。

⑥ 微分执行“PULS”指令，选择 010. 01 为脉冲发送口，选择独立工作模式，选择用 DM4 确定脉冲数。

⑦ 微分执行“SPED”指令，选择 010. 01 为脉冲发送口，选择独立工作模式，选择用 DM6 确定脉冲频率。

⑧ 如到位，延时 2 秒，作第 2 段控制，情况同第 1 段。

⑨ 起动第 2 段程序。

⑩ 复位 LR0. 00。

5.4.2　多轴协调运动控制

协调运动控制也称连续轨迹控制。在数控车、铣床，雕刻机、激光切割机、激光焊接机、激光雕刻机、快速成型机、超声焊接机、火焰切割机、等离子切割机、水射流切割机等等都有它的应用。

1. 直线圆弧插补法

（1）直线插补法

1）直线插补算法。直线插补的算法很多。这里介绍的是，基准轴输出为主，辅助轴输出配合累加插补算法。此算法使用时，起始点为当前位置，终点可按直角坐标的 X、Y 值，以脉冲为单位选定。同时，还要选定所在象限。用选定象限确定脉冲输出的方向。如第一象限，则 X 为正向，Y 也为正向。如二象限，则 X 为反向，Y 为正向。其它象限类推。在执行程序时按步计算，按步输出脉冲，而方向用方向信号控制。

所谓基准轴，就是终点坐标值较大的轴。每一步它都输出脉冲，只是当辅助轴有脉冲输出时，它先让辅助轴输出，而在后一个周期。它才输出。辅助轴是否输出，用累加的方法确定。

累加的过程是，用自身的终点坐标累加，当大于基准轴的终点坐标时，输出一个脉冲，并把累加值被基准轴的终点值减，其“差”又作为累加值存储。该算法的框图如图 5-51 所示。

图 5-52 所示为一个插补运算实例。

从图知，该例的 X 轴终点坐标为 9，而 Y 轴为 3。可知，应选 X 轴为基准轴，而 Y 为辅助轴。从框图知，一开始就累加辅助轴的终点坐标值。但它仅 3，比基准的终点坐标值 9 小，故仅基准轴输出脉冲，走第 1 步。接着，继续累加辅助轴的终点坐标值。这时累加值为 6，比基准的终点坐标值 9 小，故仅基准轴输出脉冲，走第 2 步。接着，再累加辅助轴的终点坐标值。这时为 9，不比基准的终点坐标值 9 小，故仅辅助轴输出脉冲走第 3 步。进而累计值被基准轴的终点值减，其“差”为 0，并以 0 作为新的累加值。之后如图 5-52 所示，4、5、6 又是 X 轴输出；7 是 Y 轴输出；8、9、10 又是 X 轴输出；11 是 Y 轴输出；12 又是 X 轴输出。

可知，本算法的进给的速度是恒定的，每次都只有一个脉冲。这有利于在切削加工中应用。

2）直线插补程序实现。程序是按算法编的，也是算法的具体化。其基本程序如图 5-53、图 5-54、图 5-55 所示。图 5-53 所示为步进工作起动及初始化程序。图 5-54 所示为 X 是基准轴时的脉冲输出控制程序。至于 Y 是基准轴时的脉冲输出控制程序与此类似，故不重复。图 5-55 所示为终点及方向控制程序。这里方向控制仅针对两个象限，其它两个象限略。

PLC 的步进起动程序用了基本的起、保、停逻辑。“步进起动”ON，则“步进工作”ON，并自保持。直到“步进完成”ON 其常闭触点把“步进工作”OFF。这里并联“再起动”，为了选定新的程序后重起动的需要。

这初始化程序则多是微分执行的。其功能是判断那个轴为基准轴、传送坐标值及使参数回到初始值。显然，不进行初始化，程序是无法正确执行的。

图 5-51 半累加直线插补算法框图

图 5-52 插补实例

图 5-53 步进工作起动及初始化程序

另外，还要调用 PRV 指令。读取脉冲输出状态。如图 5-53 所示，如果 W320.04 及 W310.04 均 OFF，说明两个口均已完成脉冲输出，即可调新的插补运算。至于工作节奏，可控制脉冲输出频率实现。

图 5-54 所示为当 X 是基准轴时，脉冲输出控制的梯形图程序。

如图 5-54 中“X 行程大”ON，意味着 X 为基准轴。正如算法框图指出的，它先把“y 累加值”加“y 终点”，其“和”赋值给“y 累加值”。然后，把“y 累加值”被“累加器”减，其“差”赋值给“Y_TEMP”。参看图 5-53 知，此“累加器”即为 X 轴的终点坐标值。

如不够减，即它小于“累加器”，则“y 加进位”OFF，X 轴输出脉冲。如足够减，即它大、等于“累加器”，则“y 加进位”置位。进而，把“Y_TEMP”，即这个“差”赋值给“y 累加值”，“y 步进”置位，输出脉冲。同时，把“y 已走”置位。“y 已走”置位的目的是，在这个工作周期，不让 X 轴输出脉冲。但临退出本程序段时，用“y 加进位”ON，把“y 已走”复位。为下一周期执行 X 轴发送脉冲作准备。

当下一次，再进入本程序段时，由于“y 加进位”仍为 ON，故 Y 轴的累加不进行。而使“X 步进”置位，进行 X 方向步进，同时使“y 加进位”复位，为新一次的 Y 轴累加及相应操作作准备。

以上程序讲的都是“X 行程大”ON 的情况，而“X 行程大”OFF，与此类似。在本程序最后，使“已走步数”加 1，为控制步进是否完成提供依据。

每执行一次本段程序，都会产生脉冲输出，不是使在 X 方向，就是在 Y 方向走一步。所以，执行完本程序，都将使“已走步数”加 1。

提示：在这程序中，如不用“y 已走”信号进行控制，基准轴总是输出脉冲，而辅助轴累加值超过基准轴终点值时，也输出脉冲。运动轨迹将有时是走坐标轴的平行线，有时走斜线。速度将是不均匀的。

图 5-55 所示为终点及方向控制梯形图程序。

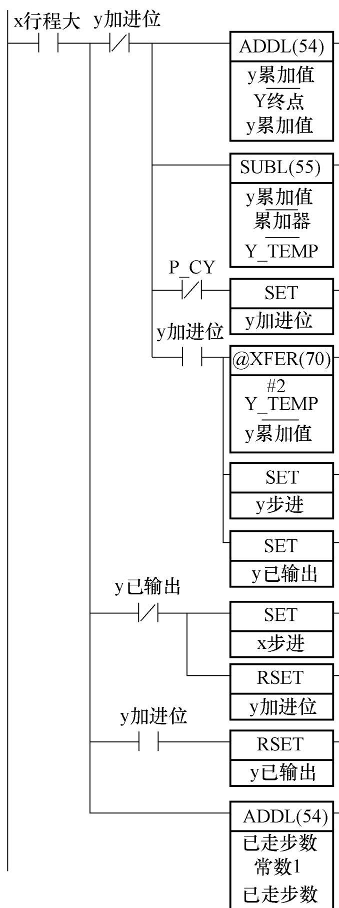


图 5-54　X 是基准轴时脉冲输出控制

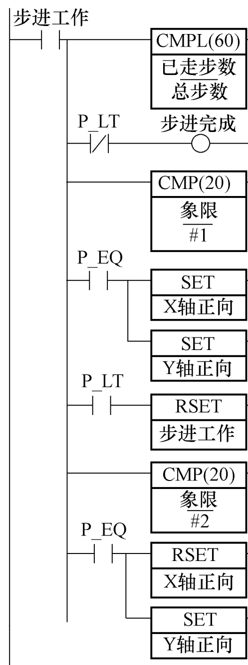


图 5-55　终点及方向控制程序

它主要是判断到了坐标终点了没有？如到，则置位“步进完成”。再看图 5-53 可知，此信号将使“步进工作”停止。

此外，此程序还依选定的象限，对 X、Y 轴运动方向做了确定。图中只列出 1、2 象限的程序，3、4 象限也类似。

（2）圆弧插补法

在两个坐标点间，如果要是曲线怎么办？把这曲线分成若干直线，用上述分几个直线段进行插补。这样，上述程序使用时及编程时，就要做分析：要分几段才能满足精度要求？每段的终点坐标是多少？然后，进行参数设定，并送入 PLC。再运行上述程序，也可实行两点间的曲线运动。早期数控实现曲线运动，加工曲面就是这个办法。结果是，使用的数据很多，编程（划分线段那样的编程）工作量很大。给数控的应用带来很大的不便。

在两个坐标点间，要走曲线第二个办法是用圆弧插补，有的还用其它二次曲线插补。这样插补比直线插补最大的优点是，达到相同的精度，所需的线段少，编程简单。但插补的运算复杂些。而后者是系统设计的问题，最终用户可不必考虑。

1）逐点比较算法。同样圆弧插补，方法也较多。这里介绍的是较常用的逐点比较法。逐点比较法的基本思想是，走一步，比较一次。也可反过来说，比较一次，走一步。比较什么？看当前的位置是在圆弧内，还是圆弧外？如在圆内，则往外走。如在圆外，则往内走。为此，插补要分象限、按运动方向进行。如在第一象限，逆时针运动：只能是正 Y 及负 X 运动，而发送正 Y 向脉冲将是往外走；发送负 X 向脉冲将是往内走。如图5-56所示。其它各象限也都有相应的规律。

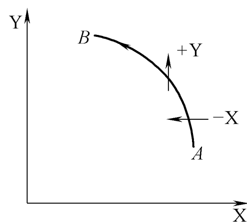


图 5-56 一象限逆时针运动

比较的关键还在于，怎么知道是在圆上、在圆内，还是在圆外？不难。如圆的半径为 R。如在圆上，则 X 的平方与 Y 的平方之“和”减去 R 的平方，即 Δ，为 0。如在圆内，则 Δ 小于 0。如在圆外，则 Δ 大于 0。

只是这里都是平方计算。但如只计算相对值，即向某方向走了一步后 Δ 变化就不用平方计算了。如现在圆上，显然以下等式成立：

$$X^2 + Y^2 - R^2 = 0$$

这时，如在正 X 向走一步，即 X 变为 X + 1，是否在圆内？可按下式计算：

$$\begin{aligned}\Delta &= (X+1)^2 + Y^2 - R^2 \\ &= X^2 + 2X + 1 + Y^2 - (X^2 + Y^2) = 2X + 1\end{aligned}$$

即只计算 2X + 1，如为正，则在圆外。同理，如正 Y 走一步则计算 2Y + 1；如负 X 走一步则计算 −2X + 1；如负 Y 走一步则计算 −2Y + 1。而这些计算没有平方，只是乘 2 及加 1，较简单。正是这样，逐点比较法用的很多。

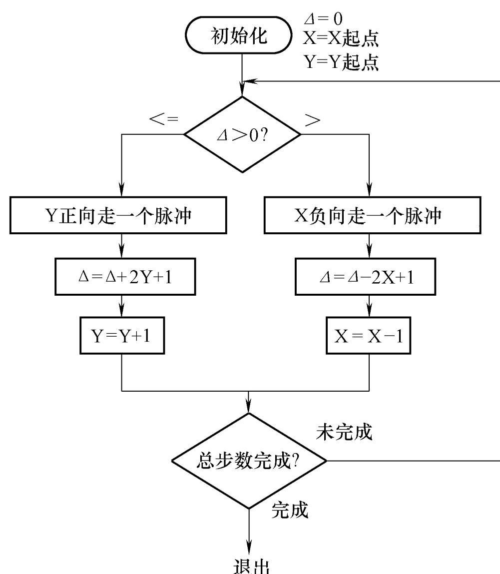


图 5-57 第一象限逆时针插补的算法

图 5-57 示的为第一象限逆时针插补时的

算法框图。

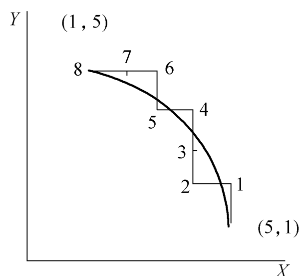


图 5-58　计算实例

从图知，先是初始化，再判断是否 $\Delta>0$？若 $\Delta>0$，则向负 X 走一步；若 $\Delta\leqslant0$，则向正 Y 走一步。再计算 Δ 及 X 或 Y 值。判断是否到终点，若到终点，则退出。若不到终点，则返回判断，重复上述过程。图 5-58 所示为用此算法计算的简单运动实例。

从图知，它用第一象限逆时针插补的算法。X 起点为 5，终点为 1。Y 起点为 1，终点为 5。其计算与走步情况如下：

第 1 次时，$\Delta=0$，走 Y，计算 Δ 为 3；

第 2 次时，$\Delta=3$，走 X，计算 Δ 为 −6；

第 3 次时，$\Delta=-6$，走 Y，计算 Δ 为 −1；

第 4 次时，$\Delta=-1$，走 Y，计算 Δ 为 6；

第 5 次时，$\Delta=6$，走 X，计算 Δ 为 −1；

第 6 次时，$\Delta=-1$，走 Y，计算 Δ 为 8；

第 7 次时，$\Delta=8$，走 X，计算 Δ 为 3；

第 8 次时，$\Delta=3$，走 X，计算 Δ 为 0，到终点，停。

提示：这里 X 起点、Y 起点是以圆心为原点计算的。X 终点、Y 终点也应是如此。即这 4 个数是有相关关系的。

2）逐点比较法程序实现。圆弧插补初始化程序如图 5-59 所示。

从图 5-59 可知，该程序先是“起、保、停”逻辑。“步进起动”ON，将使“步进工作”ON，并自保持。到“步进完成”ON，则“步进工作”OFF。这里并联“再起动”，为了选定新的程序后重起动的需要。

进入“步进工作”后，按算法进行初始化。计算总步数，把 X 起点赋值给 X，Y 起点赋值给 Y，清零“已走步数”、并把 0 赋值给“Δ”。对图 5-59 用“某大常数”赋值给，以作为 Δ 的基数。目的是使 Δ 怎么加减，也不至于出现负值（因为 CPM2A 用的是双字 BCD 码运算，出现负数不好处理）。计算后的 Δ 则与“某大常数”比较，而不与 0 比较。效果是相同的。这也算是编程因地而为的一种技巧。

图 5-60 所示为第一象限、逆时针运动插补时圆弧插补的运算程序。

从图知，它先是比较，把 Δ 与“某大常数”比较：小于或等于时，发送正 Y 脉冲，并作算法要求的相应计算；大于时，发送负 X 脉冲，并作算法要求的相应计算。

程序后面为“已走步数”加常数 1，并与“总步数”比较，如相等或大于时，则“步进完成”ON。这将使“步进工作”OFF。

图 5-60 所示程序不仅可用于第一象限、逆时针运动的插补控制，还可用于第三象限、顺时针运动。其它如图 5-61 所示，如第一象限、顺时针运动及第三象限、逆时针运动一样，在圆内，则走 +X，在圆外，则走 −Y；如第二象限、顺时针运动及第四象限、逆时针运动一样，在圆内，则走 +Y，在圆外，则走 +X；如第二象限、逆时针运动及第四象限、顺时针运动一样，在圆内，则走 −X，在圆外，则走 −Y。具体梯形图程序可在图 5-60 的基础上稍加修改即可。



图 5-59 圆弧插补初始化程序

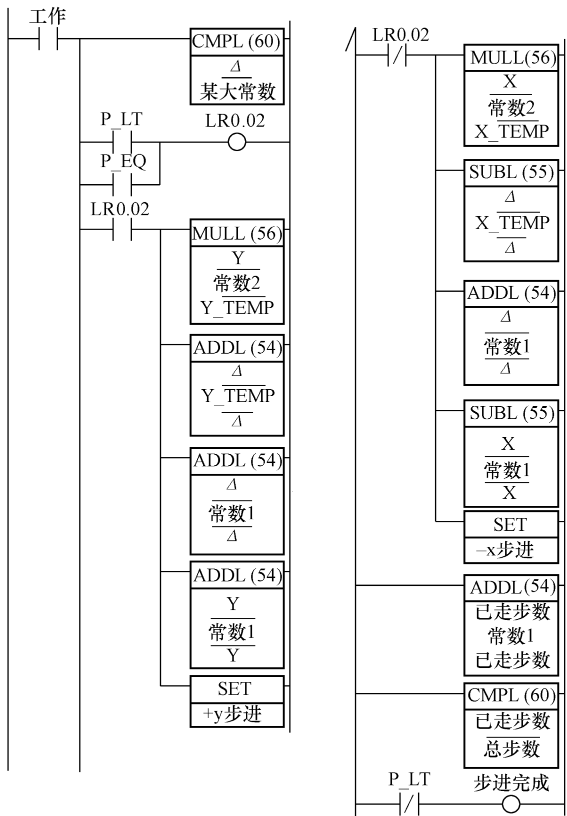


图 5-60 第一象限、逆时针运动插补时圆弧插补运算程序

（3）插补程序合成

以上介绍的程序只能做一小段运动轨迹的插补。实际曲线复杂时，要做多段插补。多段中，可能有各个象限的小直线，还可能有各个象限、逆时针或顺时针的小圆弧。怎么把这样一段段运动连贯起来，就是插补程序合成要解决的问题。

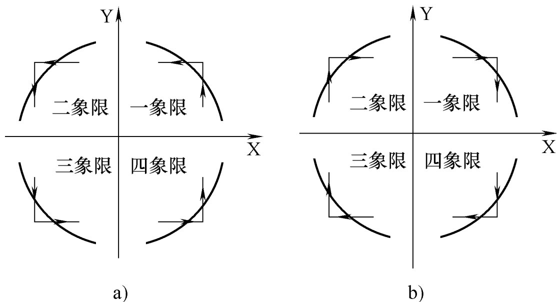


图 5-61 各象限顺、逆时针运动

图 5-62 所示是一种用程序控制实现的算法。

从图知，“程控起动”后，先传送“程控初始化数据”。这些数据有程序总数（有多少上述小段程序?）、“当前程序号”置 1、数据指针初始化（控制数据存放首地址）等。然后起动“步进工作”。“步进工作”开始时，先执行初始化程序（如图 5-47、图 5-53 所示），随后调“步工作子程序”（如图 5-47、图 5-48 或图 5-54 所示）。每执行一次，都要判断步是否完成，如未完成，则相隔一个“脉冲时间间隔”，再调此子程

序；如完成，则使“步进工作”OFF，并修改步参数指针、“当前程序号”加 1，并比较“当前程序号”是否大于“程序总数”。如不大于，则进入再起动，再使“步进工作”开始；如大于，则已完成所有程序段的控制，将结束程控。

图 5-63 所示为实现此算法的梯形图程序。

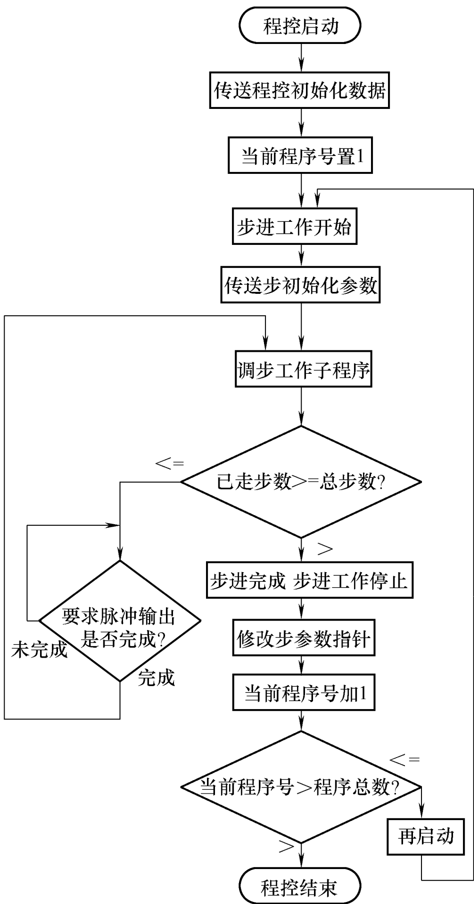


图 5-62　程序控制算法

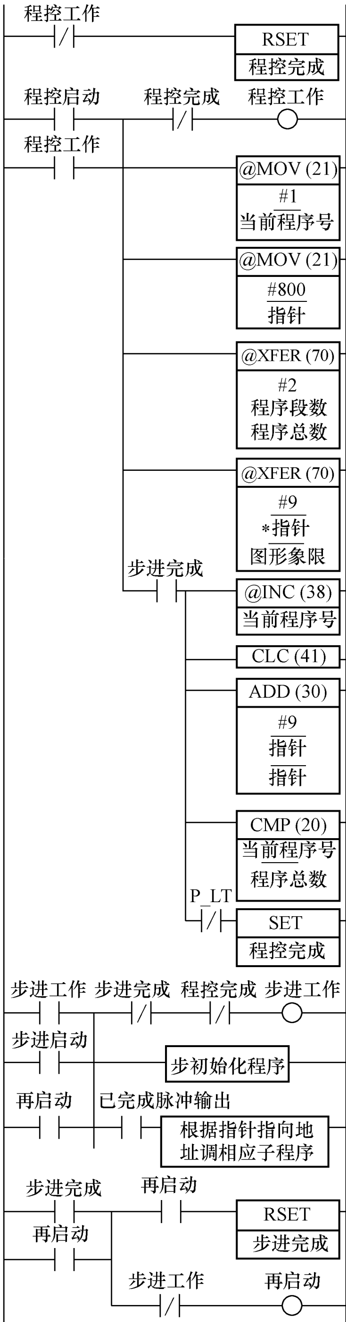


图 5-63　PLC 程序控制梯形图程序

从图知，当“程序工作”OFF 时，“程序完成”复位。当“程序起动”ON，则“程序

工作”ON，并自保持。之后进行初始化：“当前程序号”置 1，把 800 传“指针”（用作 DM 间接地址初值，即有关插补运算数据预先存储在 DM800 开始的地址中）、“程序段数”传给“程序总数”。同时，执行块传送指令，把 DM 800 开始的 9 个字数据，传送给“图形象限”等字或双字。这些数据有：“图形象限”、“X 终点”、“Y 终点”、“X 始点”、“Y 始点”。第 1 项占一个字，后 4 项每项占两字。这 9 个字是按插补要求，预先存于从 800 开始的 DM 数据区中的。有了这组数据，只要“步进工作”ON，即可定时调相应子程序，先进行步初始化（如图 5-47、图 5-53 所示），后定时调相关程序（如图 5-48、图 5-49、图 5-54 所示），即可进行相应控制。

当完成要求运动的总步数时，则“步进完成”置位。进而使“步进工作”OFF。同时，修改指针，使“当前程序号”加 1，并“指针”加 9，指向新的一组数据。而且，当“步进工作”OFF 后，从梯形图最后一个梯级知，它将使“再起动”ON，并自保持。这“再起动”ON 将重新使“步进工作”ON，并自保持。同时使“步进完成”复位。“步进工作”再次 ON，先也是步初始化，可得到新的一组“图形象限”等控制数据。进而，又可根据脉冲输出完成情况调用相应子程序，作下一个步的控制。至于程序节奏将由设定的脉冲输出频率确定。

当一段插补程序运行结束，都做“当前程序号”是否大于“程序总数”判断。如果为“大于”，则“程控完成”ON，它将使“程控工作”、“步进工作”OFF，使整个程序控制结束。

2. 目标追踪控制

上述插补算法是早期数控控制机常用的算法。随着计算机工作速度的提高及计算能力的增强，如果对象的运动轨迹有确定表达式，也可直接目标追踪实现轨迹控制，而不必作分段插补。

（1）目标追踪算法

1）根据对象运动轨迹表达式，选定一个合适的参变量，把表达式转换为参变量方程。要确保参变量的单值变化能覆盖整个运动轨迹；

2）使参变量微小增（或减）变化（变化范围应覆盖整个运动轨迹），逐一计算目标值 X、Y 的变化；

3）判断目标值 X 或 Y 是否出现单位值增、减的情况，如出现，输出脉冲，使控制对象在 X 或 Y 坐标方向作增或减 1 运动；

4）判断是否到达终点，否，则继续使参变量微小增减，重复上述 2）、3）过程；是，则控制结束。

（2）目标追踪算法实例

以下以图 5-64 所示的螺旋线轨迹为例，介绍此算法设计及其程序实现。螺旋线用参变量 θ，表示它的轨迹。

如图所示，其中 ρ 与 θ 及 X、Y 与 θ、ρ 间的关系式如下：

$$\rho = k \cdot \theta$$

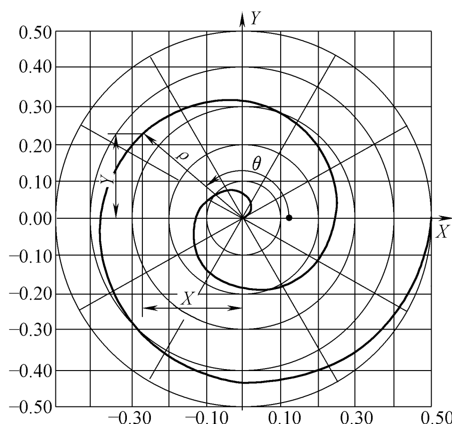


图 5-64　螺旋曲线图型

$$X = \rho \cdot \cos\theta$$
$$Y = \rho \cdot \sin\theta$$

θ 从 0 到 4π，将覆盖整个轨迹。该图表示了在 X、Y 坐标上“步进”，完成这个运动轨迹的控制情景。图 5-65 所示为这个算法图解。

从图知：

1）初始化：使目标坐标值 X、Y、实际坐标值 X0、Y0 及参变数 θ 均置 0，同时，把“某较小数”赋值给 $\Delta\theta$ 及设定螺旋圈数 n 及系数 k。

2）计算：按图示公式，计算目标值 X、Y。这里“某较小数”要小到加此值后，最多只能使 X 或 Y 的增量为 1。

3）比较：做 4 个比较，如都为“否”，则使 θ 加 $\Delta\theta$。否则如 X 与 X0 比较有一个“是”，则修改实际坐标值，作相应步进。等待脉冲时间间隔后再对 Y 与 Y0 进行比较。后者比较若“是”，则也修改实际坐标值，作相应步进。等待脉冲时间间隔后进入下一步。

4）判断：判断是否达到终点，即 Y0 = 0 及 X0 = $2\pi * n * k$ 是否成立。是退出控制，不是又使 θ 加 $\Delta\theta$，回到（2）。

可知，按图示算法，将用一段段平行于 X、Y 坐标的小短线，近似地走出螺旋线来。这个近似最大误差将不大于一个步进的行程，即脉冲当量。

（3）目标追踪算法梯形图程序实现

图 5-66 所示为实现上述算法的 PLC 梯形图程序。当未达到终点时，每隔一个脉冲间隔时间或每当完成要求的脉冲输出，将调用一次本程序。但在此没有列出工作控制及终点判断等有关程序。

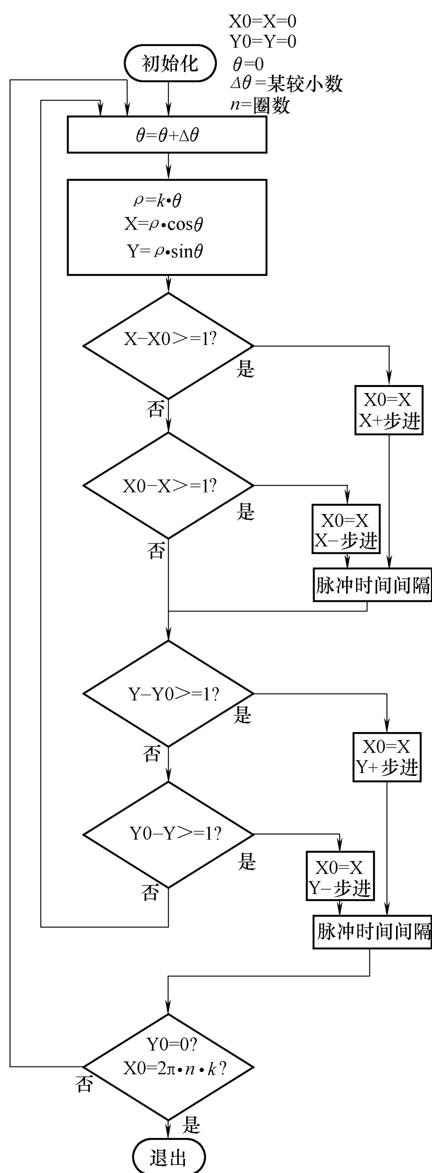


图5-65　螺旋线轨迹直接目标追踪控制算法

从图知，进入本程序时，先比较 Xi 与 X0、Yi 与 Y0 是否相等。有任一组不相等时，则 JMP 1 与 JME 之间的指令将跳过，直接进入进一步比较，确定是否设置“步进”。如满足一个设置“步进”条件，则输出脉冲，并随后传送数据，使不等的变成相等。这样，下一次再进入时，此不相等的条件将不存在了。

其后是在 FOR 、NEXT 间的指令，最多可重复执行 100 次。其任务是使 θ 增 $\Delta\theta$，按螺旋线的参数方程，进行 ρ、Xf 及 Yf 点计算。然后对 Xf、Yf 取整，并相应存于 Xi、Yi 中。接着进入比较 Xi 与 X0、Yi 与 Y0 是否相等。由于 $\Delta\theta$ 是“某较小数”。要求这个数要小到，带给 θ 变化而引起的 Xf、Yf 的增减数不能大于 1，所以，这里如不相等，也仅相差 1，可确保

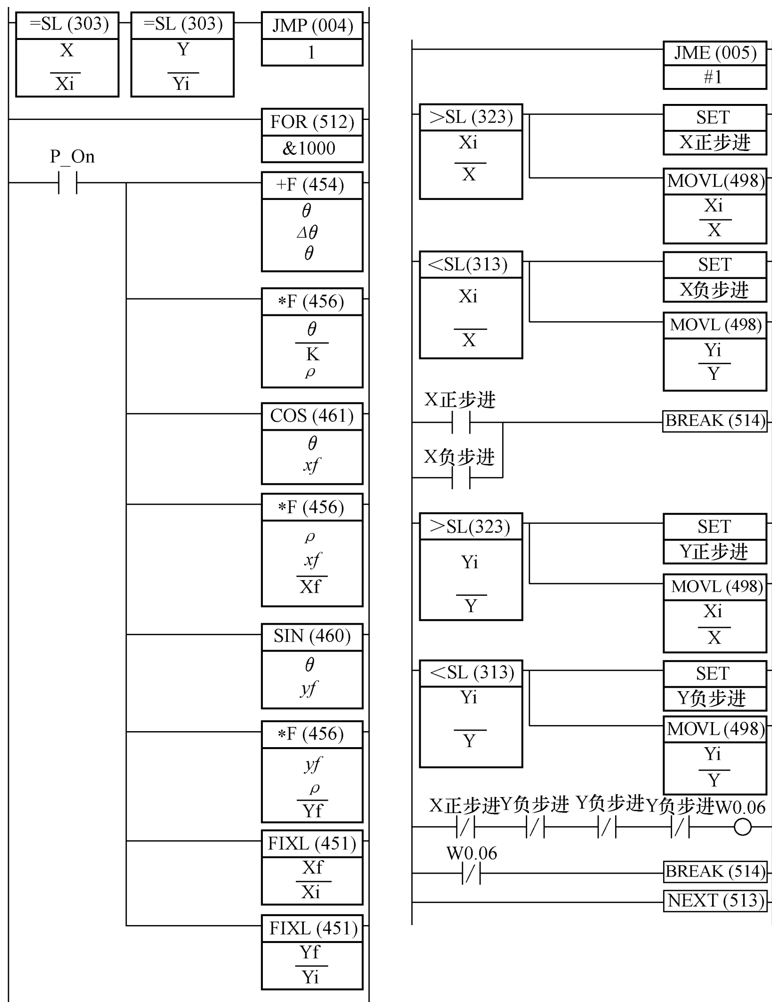


图 5-66　螺旋线运动轨迹控制梯形图程序

每次步进仅一个脉冲当量。Xi 与 X0、Yi 与 Y0 比较结果有 4 种情况：

1）Xi 与 X0 及 Yi 与 Y0 均相等，则继续重复执行，又使 θ 再增 $\Delta\theta$，及进行 ρ、Xf 及 Yf 浮点计算。然后对 Xf、Yf 取整，并相应存于 Xi、Yi 中。进而又回到这里比较。显然，由于螺旋线的 X、Y 坐标总是要随 θ 的增加而变化的，一般在 100 次内总能出现不等的情况。

2）Xi 与 X0 相等，所以，不会出现步进。而 Yi 与 Y0 不等，将使 Y 步进。并把 Yi 传送给 Y0。同时执行 BREAK 指令，跳出重复执行，跳出本程序。

3）Xi 与 X0 不等，而 Yi 与 Y0 相等，将使 X 步进。并把 Xi 传送给 X0 ，跳出重复执行，跳出本程序。

4）Xi 与 X0、Yi 与 Y0 均不等，将先使 X 步进。并把 Xi 传送给 X0 ，跳出重复执行，跳出本程序。但下一次执行本程序时，由于 Yi 与 Y0 不等，将直接进入这个比较使 Y 步进。这样处理的目的是，可做到，每次“步进”只能在一个坐标上进行，可得到均匀的运动速度。

以上讨论的螺旋线是逆时针，从第一象限开始的。还可为顺时针，或其它象限开始的。这些控制都可在本程序的基础上，稍作修改也都可实现。

（4）目标追踪算法 ST 语言功能块实现

ST 语言擅长于执行运算程序，所以用其设计本算法的程序实现是较为简单的。以下为它的功能块程序。

使用的变量声明有输入、输出及内部变量。

输入变量有：

```
xI,LINT(＊x 坐标值输入＊)
yI,LINT(＊y 坐标值输入＊)
aI,REAL(＊角度值输入＊)
dAi,REAL(＊角度增量＊)
kI,REAL(＊螺旋常数＊)
```

输出变量有：

```
x0,LINT(＊x 坐标值输出＊)
y0,LINT(＊y 坐标值输出＊)
a0,REAL(＊角度值输出＊)
xFX,BOOL(＊x 轴正向＊)
yFX,BOOL(＊y 轴正向＊)
xBJ,BOOL(＊x 步进＊)
yBJ,BOOL(＊y 步进＊)
xBS,LINT(＊x 脉冲数＊)
yBS,LINT(＊y 脉冲数＊)
```

中间变量有：

```
rN,REAL(＊螺旋半径＊)
xZn,LINT(＊x 坐标整数值＊)
yZn,LINT(＊y 坐标整数值＊)
xN,REAL(＊x 坐标实数值＊)
yN,REAL(＊y 坐标实数值＊)
```

功能块使用语句如下。其含义见语句后的注解。

```
repeat
  a0:=aI+dAi;(＊计算角度增量＊)
  rN:=kI＊a0;(＊计算螺旋半径＊)
  xN:=rN＊cos(a0);(＊计算轨迹的 x 坐标值＊)
  yN:=rN＊sin(a0);(＊计算轨迹的 y 坐标值＊)
  xZn:=REAL_TO_LINT(xN);(＊实数转换为整数＊)
  yZn:=REAL_TO_LINT(yN);(＊实数转换为整数＊)
  UNTIL(ABS(xZn-x0)>=1)OR(ABS(yZn-y0)>=1)
  (＊重复上述过程,直到 xZn 或 yZn 有大或等于 1 时停止重复＊)
  END_REPEAT;(＊重复语句结束标志＊)
  IF xZn>x0 THEN(＊判断 x 是否需要步进及确定步进方向＊)
    xBS:=xZn-x0;(＊计算 x 向输出脉冲数＊)
    xFX:=TRUE;(＊x 正向＊)
```

```
        xBJ: = TRUE;( * x 步进 * )
          x0: = xZn;( * 保留当前 x 坐标值 * )
        ELSE
          IF xZn < x0 THEN
            xBS: = x0 - xZn;( * 计算 x 向输出脉冲数 * )
            xFX: = FALSE;( * x 反向 * )
            xBJ: = TRUE;( * x 步进 * )
            x0: = xZn;( * 保留当前 x 坐标值 * )
            END_IF;
        END_IF;
        IF yZn > y0 THEN ( * 判断 y 是否需要步进及
                              确定步进方向 * )
            yBS: = yZn - y0;( * 计算 y 向输出脉冲数 * )
            yFX: = TRUE;( * y 正向 * )
            yBJ: = TRUE;( * y 步进 * )
            y0: = yZn;( * 保留当前 y 坐标值 * )
        ELSE
            IF yZn < y0 THEN( * * )
              yBS: = y0 - yZn;( * 计算 y 向输出脉冲数 * )
              yFX: = FALSE;( * y 反向 * )
              yBJ: = TRUE;( * y 步进 * )
              y0: = yZn;( * 保留当前 y 坐标值 * )
            END_IF;
        END_IF;
```

图 5-67 所示为功能块调用梯形图程序。

从图 5-67 知。它有 4 个梯形条。每梯形条的功能见注解。具体说明略。

提示：直接目标追踪控制的算法，不仅可用于本例的螺旋线轨迹控制，对于其它曲线甚至多维曲线轨迹控制，只要它的轨迹可用参数方程单值表达，也都可由本算法实现。只是要求 PLC 有很高的运算速度。



图 5-67　功能块调用梯形图程序

5.4.3　其它协调运动控制

1. 三轴协调控制

协调运动控制，除了 2 维的运动控制，理论上讲也可进行多维控制。如三维控制要控制三个坐标协调运动，其运动轨迹为空间曲线或形成曲面。当今数控机床可实现 5 坐标联动控制。

（1） 曲线轨迹。最简单的算法是目标追踪。上述螺旋线轨迹为平面曲线，如再增加一个垂直坐标 Z，用的参变量也是 θ，如 Z 与 θ 的关

系为

$$Z = kz * \theta$$

则将可实现对立体螺旋线的轨迹运动。

还是一句话，只要能列写参变量表达式，什么轨迹的运动控制，均可实现。

（2）曲面形成。常用的形成是用“行切法”。办法是，把曲面与平行于坐标面的平面（如平行于 X、Y 坐标面的平面）相截。选取不同 Z 值，将生成不同的平面曲线。如能弄清这些平面曲线的解析式，则可用目标追踪控制在这个平面上的 X、Y 运动，以形成所要求的轨迹。如 Z 值能在其定义域内依次全部选取，则这些轨迹的叠加，即可形成所要求的曲面。

当然这里的程序设计是困难的。而且控制速度与控制规模的矛盾将更加突出。所以，较高性能的运动控制还是使用专用的硬件模块为好。

2. 相对值数据列表运动控制

除了插补、目标跟踪，还可用数表，即数据列表，进行运动控制。

（1）算法要点。把运动轨迹按单位脉冲分步，依次计算每步的 X、Y（取决于坐标数）增量值，并列写成数表，存储于 PLC 的数据存储区（D 区）中。PLC 程序就是按数表要求，向各个坐标输出脉冲，进而实现协调运动。

这对既不便进行直线或圆弧插补，又没有合适的解析式可用的运动轨迹控制是惟一可实现的算法。事实上，早期，如航空发动机涡轮叶片这样的型面的数控加工就是这么处理的。

（2）实现程序。为了节省内存，该程序用的数表为每个字的 4 个数位，分别存一组数据，即 X 及 Y 两坐标的增 1 或减 1 值。其含义是，数位 0：字的 0 位 1，表示 X 增 1，1 位 1，表示 X 减 1；字的 2 位 1，表示 Y 增 1，3 位 1，表示 Y 减 1；数位 1：字的 4 位 1，表示 X 增 1，5 位 1，表示 X 减 1；字的 6 位 1，表示 Y 增 1，7 位 1，表示 Y 减 1。数位 2、3 依次类推。所有数据存储在 D0 开始的字中。CP1H 机 D 区有 32k 字，可存储 128k 组数据。每组数据若生成一个脉冲，而脉冲当量若为 0.01 毫米，则可控制 1280 毫米的行程。这样行程与分辨率对一般机床是足够用的。图 5-68 所示为这样增量值数表使用程序。

该图有 16 个梯形条。条 1 为起动与初始化。起动使 W0.10 ON 后，把 D0 的地址赋值给索引寄存器 IR0，并使 W98、W99（用于计数）、W80、W81（用于数位移位处理）清零。同时，调 PRV 指令，读取脉冲输出口 0、1 的状态。如果已起动并两个输出口已完成脉冲输出，将按“ON”条件执行梯形条 3 到 16 间指令。

梯形条 14 用于计数及使数位变化。当每次脉冲输出完成，都使 W98、W99 加 1，同时把 W98、W99 被 4 整除。除后，其商存于 W79、W80，余数存于 W81、W82。显然，随着 W98、W99 值的变化，W81 的值将在 0 到 3 之间变化。这样，W98、W99 可用于结束控制。而 W81.00 与 W81.01 的变化，即可用于控制 4 个数位的使用。

梯形条 3。由于开始时 W81 为 0，即 W81.00 与 W81.01 常闭触点均 ON，故可把 IR0 指向的地址字（开始时为 D0）的值赋值给字 W97。然后 IR0 加 1（修改指针），为下一次调用（开始之后为 D1，以后依次类推）作准备。注意，W97 只在 W81.00 与 W81.01 常闭触点均 ON 才赋值。所以，赋值一次之后，得到了再次出现 W81.00 与 W81.01 常闭触点均 ON 的条件才作新的赋值。这时，正好完成了 4 个数位的使用。

梯形条 4、5、6、7 执行数位（digit）传送指令 MOVD，将根据 W81.00 与 W81.01 的变化把 W97 的 0、1、2、3 数位传送给 W100 的数位 0。梯形条 8、9、10、11、12、13 用于处理 x、y 坐标方向及输出，它的依据就是 W101 数位 0 的值。而实际就是预存于 D0 开始的数表。

梯形条 15 为终点控制。到了数表结尾将复位 W0.10，使工作（W0.10 OFF）停止。图中设定终点数位数（存储字数乘 4）为 64000，实际可按需要更改。

（3）数表生成。可在计算机上用仿真程序生成，生成后下载给 PLC。也可使用 PLC 程序自身生成。这样，PLC 使用数表时，可简化计算，提高控制速度。还可人工填入。这对不规则的运动协调控制可能是惟一的方法。好在这些数据可以存储，开头麻烦些，但填入可长期使用。

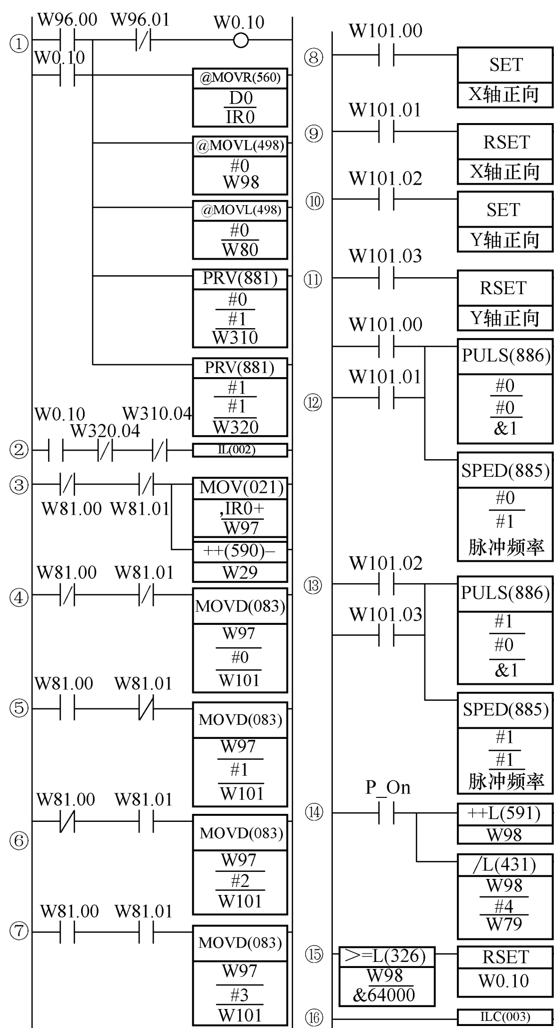


图 5-68　增量值数表运动控制程序

3. 绝对值数据列表运动控制

除了相对值数据列表，还可用绝对值数据列表进行运动控制。

（1）算法要点。把运动轨迹各个坐标的绝对值按步，列写成数表，并存储于 PLC 的数据存储区（D 区）中。PLC 程序就是按步，逐次计算数表中相邻“步”之间的差值。并根据差值的符号及大小，确定各个坐标输出脉冲数及输出方向，进而实现运动的协调运动。

（2）实现程序。图 5-69 所示为这种绝对值数表运动控制程序。该图有 17 个梯形条。条 1 为起动与初始化。起动使 W5.10 ON 后，把 D100 的地址赋值给索引寄存器 IR0 及 D101 的地址赋值给索引寄存器 IR1，并使 W25（用于计步数）、W60、W61、W62、W63、W80、W81、W82、W83（用于计算）清零。同时，调 PRV 指令，读取脉冲输出口 0、1 的状态。如果已起动并两个输出口已完成脉冲输出，将按“ON”条件执行梯形条 3 到 17 间指令。

梯形条 2，执行 IL 指令及把 IR0 及 IR1 指向的 D 字值传送给 W50（当前 X 坐标值）、W51（当前 Y 坐标值），并使 IR0 及 IR1 的值加 2（为读取下一坐标值作准备）。这说明 D100 开始的偶数地址存储的为 X 坐标值，而奇数地址存储的为 Y 坐标值。

梯形条 3、4、5、6 用以处理 X 坐标。先是判断下一坐标值是大还是小于当前坐标值。如下一坐标值大，则置“X 轴正向”ON，反之置其 OFF。同时计算两者的差值，并乘以“计

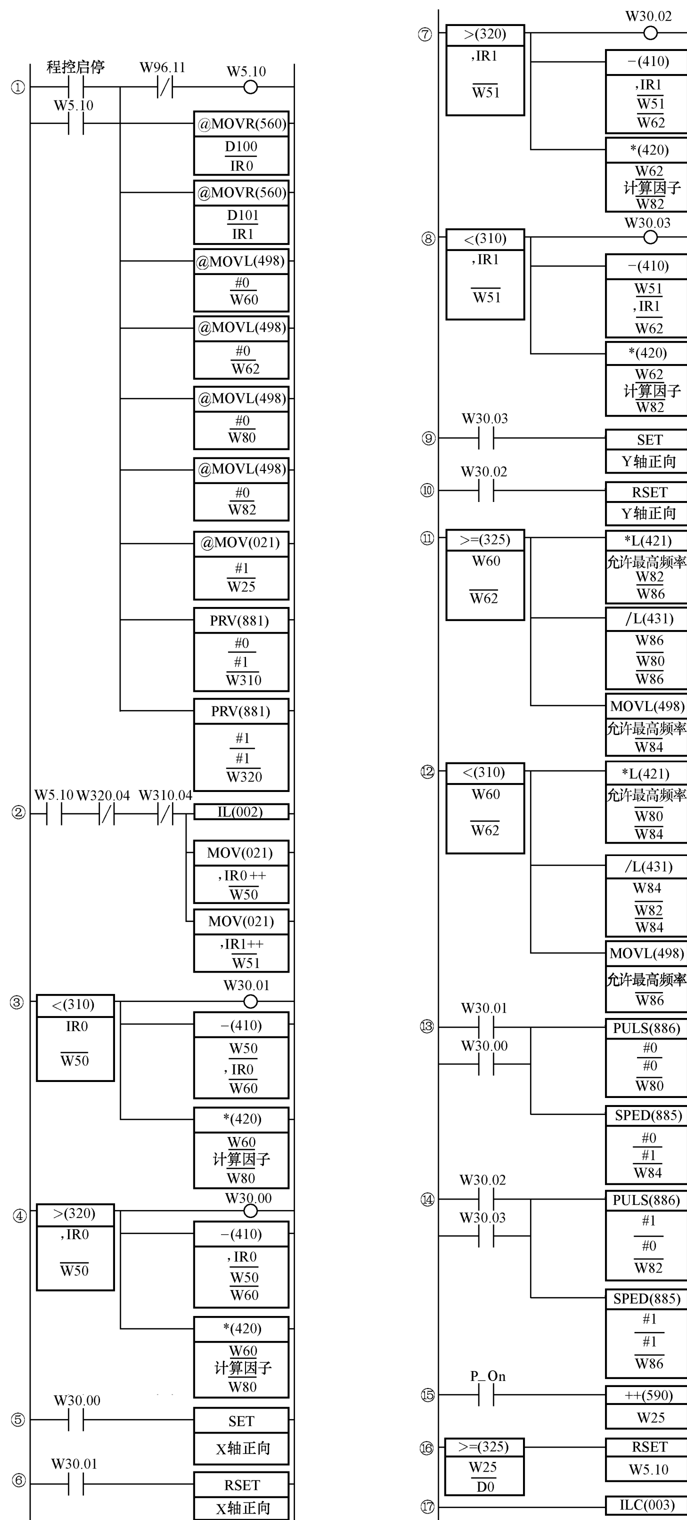


图 5-69　绝对值数表运动控制程序

算因子”，确定应该输出的脉冲数。这里的“计算因子”是用以协调数出脉冲数与实际位移的关系。梯形条 7、8、9、10 用以处理 Y 坐标。情况类似。

梯形条 11、12、13、14 用以处理输出脉冲频率及实际输出脉冲。先是比较两个坐标输出脉冲数，然后把“允许最高频率”赋值给脉冲数多的坐标输出口，再按脉冲数比例计算脉冲数小的坐标的输出频率。确保两个坐标输出脉冲频率与输出脉冲数成比例，而且最大脉冲频率不超过允许值。

梯形条 15 用于计数。当每次脉冲输出完成，都使 W25 加 1。梯形条 165 为终点控制。到了数表结尾将复位 W5.10，使工作（W5.10 OFF）停止。图中设定终点步数存于 D0 中，实际可按需要更改。

（3）数表生成。可在计算机上用仿真程序生成，生成后下载给 PLC。也还可人工填入。好在这些数据可以存储，开头麻烦些，但填入可长期使用。用绝对值数表的优点是，如果是直线运动，填入起点及终点值即可，中间靠输出脉冲数及相应的频率保证。当然，它的精度不如相对值数表。

4. 虚拟轴运动控制

1965 年，Stewart 提出著名的用于安装天文望远镜的 Stewart 平台机构。并联机床（Parallel Machine Tool，简称 PMT），也被称为虚（拟）轴机床（Virtual Axis Machine Tool），是基于 Stewart 并联机构的研究而发展起来的。它在 1994 年美国芝加哥机床展上首次面世。

图 5-70a 所示为一种并联机床的内部结构。在这里的多棱体桁架的 5 个面上，安装有滚珠丝杆的支点——万向铰链，5 根丝杠的另一端通过铰链与主轴部件的 5 个可转动同心外环连接。改变 2 个支点间的距离，即可使主轴部件处于不同工作位置。5 杆并联机构驱动的主轴部如图 5-70b 所示。

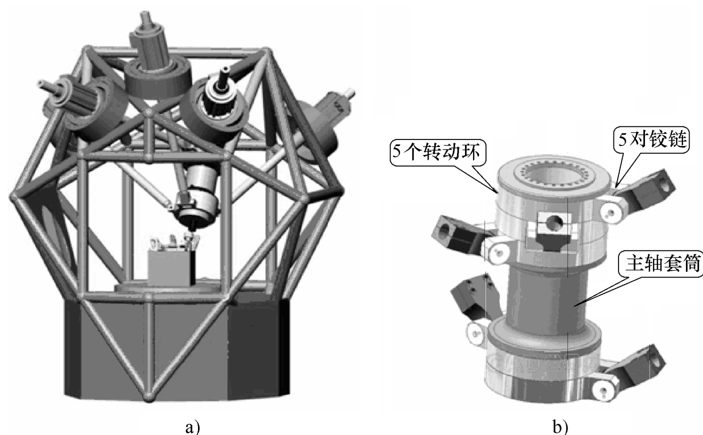


图 5-70 并联机床结构及主轴部件

a）并联机床结构 b）并联机构驱动的主轴部件

这种新型机床结构合理，运动部件磨损小。具有高刚度、高承载能力、高速度、高精度以及重量轻、机械结构简单、制造成本低、标准化程度高等优点。还适合于模块化生产。对于不同的机器加工范围，只需改变连杆长度和接点位置。维护也容易，无须进行机件的再制和调整，只需改变机构的参数。

此外，还有由并联、串联同时组成的混联式数控机床，不但具有并联机床的优点，而且

在使用上更具实用价值。类似的，也还有并联工作的机械手。也是通过虚拟轴的伸缩达到控制运动轨迹或目标位置的目的。

由于此类机床或机械手没有实体坐标系，其设计与运行要用到复杂的数学计算与推理，即所谓要“用数学制造机床”。所以，它的坐标系与工件坐标系的转换都要靠软件完成，其控制也要用计算机实现。相信随着这种机床与机械手应用的普及以及 PLC 运动控制技术的进步，利用 PLC 对其实施控制也将是可能的。因此对此先作些了解也是必要的。

5.4.4　运动控制细节处理

设计运动控制程序还有一些细节需要正确处理。没有处理好将难以实现预想的目标。这些细节有：

1. 累计误差

使用相对坐标的脉冲控制，每循环运行一次，有可能回不到原始位置，出现微小误差。反复多次运行，将使误差积累，以至超出控制精度的要求。为此，每次循环运行后最好能进行一次基准点的校正。

校正的方法是，使用精密行程开关监测基准点附近位置。当从一个预定的方向运动到此，它连接的输入点 ON（或 OFF）时，再继续前行，再到旋转编码器复位输入 ON 时，这时的位置就是基准点。这时，可激活运动初始化运动程序。这样，每次循环运行生成的误差，都将消除，不会积累。长期工作也可确保控制精度。

如果使用绝对坐标系统，则不必每个循环运行都做校正。只需开机时校正就可以了。

2. 间隙消除

由于机械原因，运动系统间隙是不可避免的。这样一来，当运动变向时，将可能丢步。为此，完善的运动控制都有相应的间隙补偿手段。

补偿可以用硬件消除间隙实现，但完全没有间隙难以做到。最实际的还是用软件实现。办法是，在运动方向开始改变的时刻，多输出若干脉冲，先消除间隙，然后再正常反向运动。图 5-71 所示是一个程序实例。

图中梯形条 3、4 作为本周期输出方向的记录。如果方向改变的首次输出，如本例 X 轴方向改变，其输出脉冲将从正常的 1 个变为 5 个。当然，这个改变数，应根据实际间隙调整。

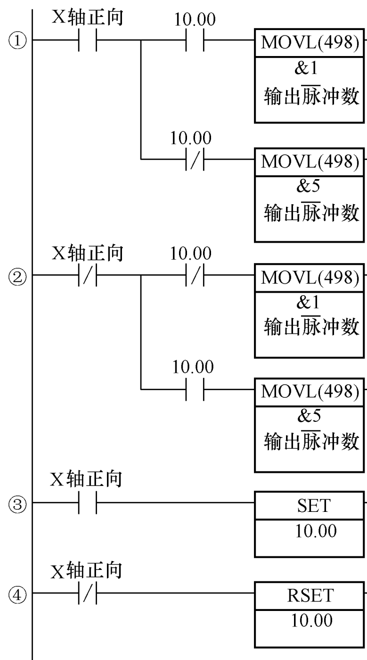


图 5-71　间隙补偿程序实例

3. 手动调整

运动控制不仅要有正常的工作程序，还要有调整程序。前者多少联动，可自动工作。后者多手动，且功能单一。虽较简单，但也是必不可少。如上述基准点的处理，多用手动处理。

4. 安全控制

运动控制系统工作速度较大，如果控制不当或系统失灵，容易出现安全问题。所以，对系统控制可能

出现的安全异常，要心中有数，并通过附加程序予以控制。这些异常有：

（1）极限位置控制。运动部件不能在一个方向无限制的运动，在其极限一般要放置限位开关。一旦部件超出此位置，应控制其停止运动。或利用脉冲输出状态对运动控制做互锁。

（2）失步控制。使用步进电动机进行开环控制，有可能失步。对此，必要时也有措施。如使用步进电动机，脉冲频率不能超过允许值，并要保证脉冲发送完成后，再作新的输出。

（3）互锁。有时有多个部件，其运动是相互制约的。这时，就要按约束条件建立工作互锁，以确保部件工作安全。即使同一部件不同方向运动，必要时也要互锁。这在实际电气控制中是很常见的。PLC 程序没有处理也可用电气系统处理。

5.5 同步控制

关键词：速度同步控制、位置同步控制

同步控制可用闭环控制，其原理与本章的 5.3 节基本相同。只是它以被追踪的量作为闭环控制的给定值。也可用开环控制，只监测被追踪量，以使控制输出能随追踪量的变化而变化。前者较复杂。后者使用脉冲量实现控制，既简单，又能保证控制精度。

同步控制在套色印刷、包装机械、纺织机械、飞剪、拉丝机、造纸机械、钢板展平、钢板延压、纵剪分条等都有应用。

5.5.1 开环同步控制

本书第 4 章 4.9.4 节介绍传送带的模糊控制，实际也是速度同步的控制。以下介绍的是使用脉冲技术实现同步控制。

1. 速度同步控制

要求输出的脉冲频率保持与输入的脉冲频率一致。办法是先读入脉冲频率，然后按读入的频率起动输出。其程序如图 5-72 所示。

这里只有两个工作指令。一是 PRV，读跟踪的输入脉冲频率，并测得的频率存储于 D0 中。另一是 SPED，按跟踪得知的频率，D0 值，连续输出脉冲。

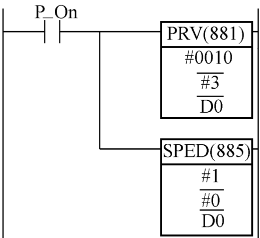


图 5-72 程序跟踪

2. 位置同步控制

（1）同时输出脉冲。同时向目标对象与跟踪对象输出相同的脉冲，靠脉冲驱动系统实现同步。这样的同步控制其实就是单一控制的重复。开环直线插补实质上就是开环位置同步控制。不同的只是它的位置移动比例可以调整。

（2）程序同步。边测量目标对象位置，边向跟踪输出脉冲，以达到两个运动同步。图 5-73 所示为一个同步处理程序。

这个程序每间隔一定时间采集一次检测到的目标运动时输入脉冲现值。D0、D1 用一存储采集反映目标位置的脉冲输入值。D10、11 用于暂存此值。用这两个比较，确定控制对象运动方向及输出的脉冲数，从而消除两者位置的偏差。只是，这样的位置跟踪在时间上是存在滞后的。但对目标对象另有自己的驱动系统也只好这么处理。

（3）指令跟踪。CPM2A 机有（SYNC）指令，执行它，可使脉冲输出与采集到的脉冲

输入成一定比例。此指令梯形图格式如下：

@SYNC(—)	@SYNC(—)
P1	P1
P2	P2
C	C

这里的 P1 为指定脉冲输入口，如设为 000，指定点 00000 为脉冲输入；P2 为指定脉冲输出口，如 000 指定 01000 点输出，如 010 指定 01001 点输出；C 为同步因子，BCD 码，占一个字。取值为 0～1000。其含义为输出脉冲频率与输入脉冲频率之比。

只是，本指令为扩展指令，使用前要指定功能号，并要下载给 PLC。在下载前，还要把 PLC 设置成容许扩展指令功能码下载模式。办法是使 DM6602 的高字节设为 1，否则无法下载。同时，对 DM 6642 的 08～15 位也要作相应设定。否则本指令无法正确执行。其设定要求如表 5-18 所示。

3. 应用实例

在 2008 北京奥运会上使用的撑杆跳杆的升降与前后移动，就是典型的双轴位置开环同步控制。图 5-74 所示为它的使用现场俄罗斯女子跳高名将伊辛巴耶娃，正在破 5 米 05 世界纪录摘金的激动场面。

从 5-72 图知，它有两个跳杆立柱，其上各有一个支架。跳杆可在这支架上前后移动。而支架又可沿立柱升降。显然，为了跳杆能正确工作，这两组的移动与升降必须同步。

图 5-73　同步位置开环跟踪程序

表 5-18　DM 6642 设定

DM 6642	00～03	高速计数模式 0：两相输入（5Hz） 1：脉冲＋方向（20Hz） 2：增减模式（20Hz） 4：增模式（20Hz）
	04～07	高速计数复位模式 0：硬件＋软件复位；1：软件复位
	08～15	对 00000 到 00002 高速计数及同步控制设定 00：不使用 01：使用高速计数 02：同步脉冲控制（10 到 50Hz） 03：同步脉冲控制（20 到 1Hz） 04：同步脉冲控制（300 到 20Hz）

该系统机械部分由泰山体育产业集团生产，运动控制部分由沈阳旭风电子科技开发有限公司研制。系统控制机使用 OMRON CP1H PLC，操作界面使用 OMRON 5.7in 人机界面，伺服系统用松下伺服电动机（含驱动器）及 ABBA 步进电动机（含驱动器）。

图 5-75 所示为它的人机界面的一个操作画面。设置好预定高度，再按下触摸屏上的“起动高度”按钮，即可使支架上升到预定的设置高度。设置好预定的架距，再按下触摸屏上的“起动架距”按钮，即可使跳杆前移到设置的架距。按下触摸屏上的“回原点”按钮，则可使支架、跳杆返回到原始位置。



图 5-74　撑杆跳杆控制系统使用现场



图 5-75　操作系统画面

该系统除了处理好硬件，还要编写 PLC 及人机界面程序。人机界面程序可使用包装在 OMRON 的 CX-one 软件中的 CX-Designer 软件在计算机上编写。编写后下载给人机界面。这类程序的编写将在本书第 7 章作进一步介绍。

PLC 程序要点有，数据处理（考虑传动机构参数，把设定数据转换为脉冲输出）、微调系统（微量移动）、工作输出（根据工作命令输出脉冲）、原点设置（确定基准点）、手动设置（接受人工设定数据输入）、伺服报警（出现运动异常时提示）等。其同步控制较简单，靠两组相同的开环脉冲输出实现。

经过多场奥运大赛证明，这套系统不仅使用方便、工作可靠，而且跳杆的升降速度达 0.8 米每秒，为国内外同类产品的 4 倍。深受运动员、裁判员的好评。为北京实现科技奥运的承诺作了贡献。

提示：高性能的伺服电动机也有位置跟踪功能。也可用以实现位置同步。

5.5.2　闭环同步控制

闭环同步控制与一般闭环控制不同的只是，它的设定值是由测定的跟踪对象确定。所以，它要有两个检测系统。一是检测跟踪目标的位置值，另一是检测控制输出的反馈值。闭环同步控制虽较复杂，但对目标对象另有自己的驱动系统，要求又较高时，还是要用到的。以下以某导弹光学瞄准器的仰角，追踪雷达仰角的例子，来说明这个闭环同步控制。

由于结构的原因，光学瞄准器与雷达天线无法同轴安装，造成当目标距离不同（影响较小）、仰角不同时，两者测得的仰角数据略有差别。为此，须按仰角进行人工修正。

为此，该系统用一个绝对旋转编码器检测雷达仰角，用另一个绝对旋转编码器检测光学瞄准器仰角。雷达与光学瞄准器一起有自身的驱动。但光学瞄准器还另用步进电动机驱动，可与雷达相对运动，以进行角度修正。

由于角度修正值与仰角间的关系不是线性的，用公式计算也较麻烦，故先用手动操作，建立仰角与修正值之间的关系数表。

运行程序时，由绝对旋转编码器读入雷达的仰角，再根据仰角查此数表，求得光学瞄准器的仰角修正值，并以此做闭环控制的设定值。闭环控制的目的是使检测到的光学瞄准器仰角与修正值相等。在这种情况下，步进电动机仅是一个运动源，不进行运动角度控制。该系统程序是沈阳鹭岛公司工控部开发的，具体内容略。

5.6　特殊单元控制

关键词：位置控制单元、运动控制单元、运动控制器

运动控制用的特殊单元有位置控制单元及运动控制单元。

位置控制单元用于运动部件的位置及速度控制。可指定若干目标位置，并控制其在若干目标位置间按一定规律运动。这些规律有起动速度、加速度、运行速度、减速度、结束速度等。按坐标分，有单轴的，还有双轴的、多轴的。双轴的、多轴的可单独控制，又可协调控制（但只能直线插补）。

运动控制单元不仅可进行两个轴的位置控制，而且还可实现两轴间直线插补及圆弧插补等协调运动。还可用 G 语言编程，而无需梯形图；可存储的程序也多，操作更方便。

此外，OMRON 还推出灵活快速运动控制器（Flexible Quick Motion ，FQM），专用于多轴运动控制。

5.6.1　位置控制单元

OMRON 位置控制单元与大、中型 PLC 配套使用。如中型 PLC 用的，有 C200H 的 NC112、NC211、NC113、NC213、NC413、C200HWNC413/213/113 及 CJ1W 的 NC113、213、413、133、233、433，等等。这些单元接受 PLC 指令，或自身存储的数据，向电动机驱动器输出脉冲信号，分别进行单轴控制、双轴控制或四轴的定位控制。以下主要针对 CJ1 机的情况作简要介绍。

1. 结构

位置控制单元也称 P(POSITION)C(CONTROL)U(U-NITE)。图 5-76 所示为 CJ1W-NC413 的外观图。

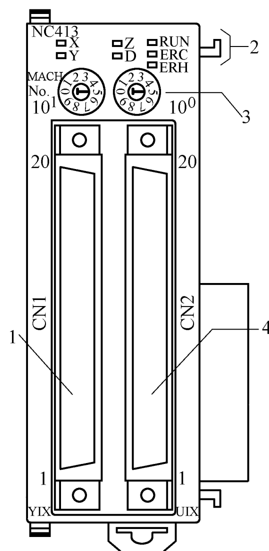


图 5-76　CJ1W-NC413 外观

1—X/Y 轴连接器连接到步进电动机驱动器或者伺服驱动器（控制两个轴）　2—LED 指示灯显示 PCU 的操作状态　3—单元数设置开关设置 PCU 的单元号　4—Z/U 轴连接器连接到步进电动机驱动器或者伺服驱动器（ 控制两个轴）

上述所有位控单元的基本原理如图 5-77 所示 。

从图 5-77 可知，位置控制单元有自己的微处理器(MPU)、内存、总线。同时，还有脉冲发生器及 I/O 接口。

它一方面通过 PLC 总线接口，与 PLC 的 CPU 相连；另一方面又通过 I/O 连接器输出脉冲及接受少量的开关量输入（图中外部输入）。这样，既可接受 PLC 的有关控制命令，并存于自身的内存中，又可接受开关量输入信号，如极限位置信号、原点信号等，以便自身的 CPU 进行处理，并依处理结果控制脉冲发生器输出脉冲。同时，在 PLC I/O 刷新时，还反馈一些信息给 PLC，及接收来自 PLC 的新

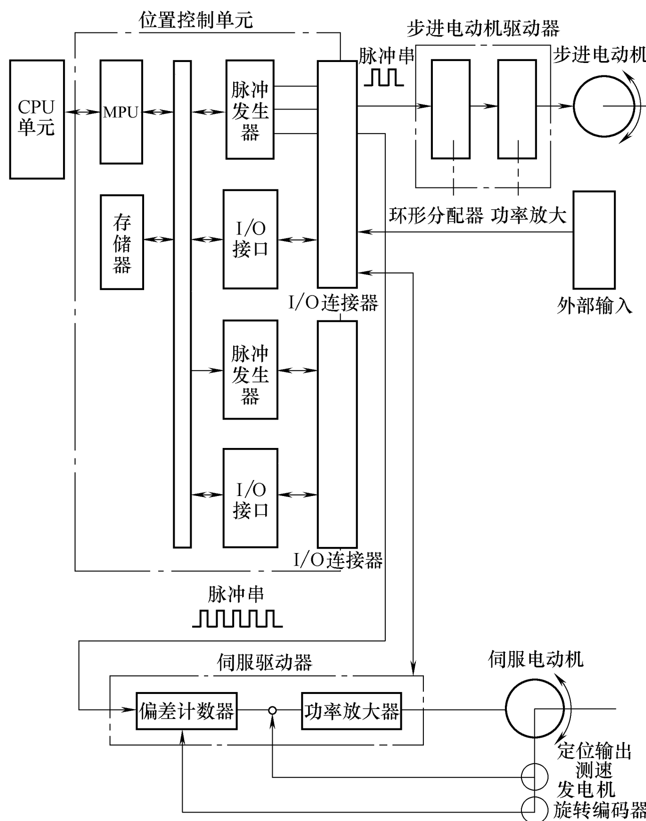


图 5-77 单轴的位控模块的功能框图

命令。

它的脉冲输出可有两种形式：一为脉冲信号加方向信号，如图 5-78a 所示；另一为两路脉冲信号，即正向（增）脉冲及反向（减）脉冲，如图 5-78b 所示。各有其信号线。伺服系统为开环时用前者，而闭环、半闭环时用后者。到底使用什么样的脉冲信号，可由单元的 DIP 开关设置确定。

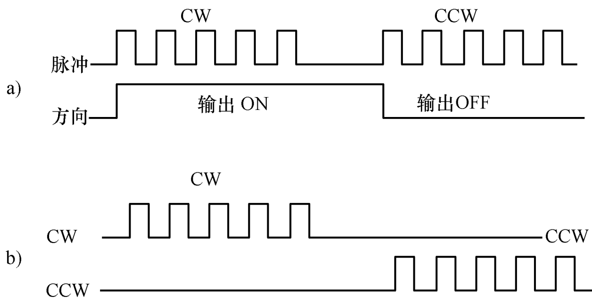


图 5-78 两种形式脉冲

a）脉冲信号加方向信号 b）两路脉冲信号

图 5-77 画了两种伺服系统，其实用其中的一个就可以了。而且也只能用一个。从图 5-77 可知，开环时，其后要接环形分配器，把连续的单相脉冲信号分成多相，并依方向信号确定其相序，以控制电动机转动方向。多相脉冲信号经功率放大，再去驱动步进电动机。步进电动机的运动经机械传动，再带动部件运动。开环系统较简单，但功率不大。

图 5-77 还示出闭环系统。它有一个偏差计数器。这计数器既接收来自位控单元的正向与反向脉冲，又接收来安在运动部件上的旋转编码器的反馈脉冲。这几股脉冲综合后，由偏差计数器产生相应输出。这输出放大后，再去驱动伺服电动机。显然，只要有偏差，即给定脉冲不等于反馈脉冲，伺服电动机就有速度。直到无偏差，即达到输出脉冲要求的位置，伺服电动机才停止转动。该图还接了一个速度反馈回路，目的是使部件运动速度平稳，避免振荡。

图 5-79 所示为 CJ1W-NC413 位控系统配置情况。

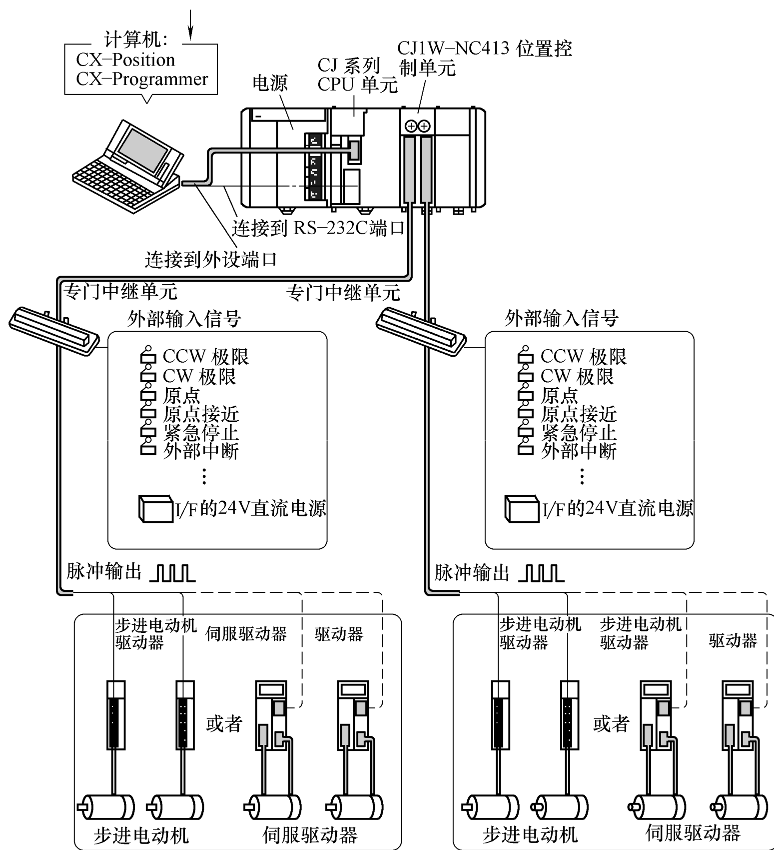


图 5-79　位控系统接线及安装

从图知，其输入的开关量信号有：CCW（逆时针转动极限位置）、CW（顺时针转动极限位置）、原点、原点接近、紧急停止、外部中断等。输出为脉冲信号，可接步进电动机，也可接伺服驱动器（图中虚线所示）。该模块为 4 轴，故可接 4 套电动机。

2. 性能

高速响应定位控制，PCU 对于来自 CPU 单元的指令可在 2ms 内做出响应。

定位控制有两种方法：第一种是存储器操作，在这种操作中定位所需的信息先传输到 PCU，然后由 PCU 实施控制；第二种是直接操作，在这种操作中每次的目标位置和目标速度都要由 CPU 单元设置。

定位可以在 -1073741823 ~ 1073741823 个脉冲位置范围内被执行，速度在每 1 个脉冲单位中在 1 ~ 500000p/s 范围内。这意味着在大范围内以精确的速度定位是可能的。

与基于 Windows 的 NC 支持软件（CX-Position）兼容，可用其在 Windows 环境下设置 PCU。

CJ1W 的几个型号的主要性能如表 5-19 所示。

由表 5-19 可知，这里特性参数多与脉冲有关。而脉冲与运动有什么关系呢？这可用脉冲当量 P 表达：

$$P = t \cdot \theta / (360 \cdot i)$$

这里　P——脉冲当量，每脉冲运动部件的移动量，mm；

t——传动丝杠的螺距，mm；

θ——每脉冲电动机转动角度；

i——传动比，电动机与丝杠转速之比。

表 5-19 性能

项		型　　号		
		CJ1W-NC 113/133	CJ1W-NC 213/233	CJ1W-NC 413/433
控制轴数目		1 个轴	2 个轴	4 个轴
定位操作	两种类型:存储器操作和直接操作			
	独立	1 个轴	2 个独立轴	4 个独立轴
	线性插补	没有	最多 2 个轴	最多 4 个轴
	速度控制	1 个轴	2 个独立轴	4 个独立轴
	中断进给	1 个轴	2 个独立轴	4 个独立轴
位置	范围	-1073741823 ~ 1073741823 脉冲(见注释)		
	数据项	100/轴		
速度	范围	1 ~ 500000p/s		
	数据项	100/轴		
加速和减速时间个数	范围	0 到 250 秒,直到达到最大速度		
	数据项	对于加速和减速分别 9/轴		
功能和设置	原点搜寻	原点接近输入信号:可选(无,N. O. 或者 N. C. 接点) 原点输入信号:可选(N. O. 或者 N. C. 接点) 原点补偿:-1073741823 ~ 1073741823 个脉冲 原点搜寻速度:可以设置高速或者接近速度 原点检测方法:可以设置成渐进输入信号转成 ON(打开)后在原点输入信号上停止,渐进输入信号转成 OFF(关闭)后在原点输入信号上停止,不使用渐进输入信号在原点输入信号上停止。或者限制输入信号转成 OFF(关闭)后在原点输入信号上停止。N. O. = 常开　N. C. = 常闭		
	点动	使部件先以一定速度运行,然后停止		
	停顿时间个数	19 次/轴,可以设置为从 0 到 9.99 秒(0.01 秒为单位)		
	加速/减速曲线	梯形或者 S 曲线(可以对不同的轴分别设置)		
	区域	当当前位置在一个的区域中时,区域标志转为 ON(打开)。可以为每个轴设置 3 个区域		
	软件限位	可以在 -1073741823 ~ 1073741823 脉冲范围内设置		
	间隙补偿	0 ~ 9,999 脉冲,也可以设定补偿速度		
	示教	用来自 PLC 的指令,的当前位置可以被当作位置数据		
	减速停止	STOP(停止)命令使得定位根据的减速时间减速至停止		
	紧急停止	脉冲输出被外部紧急停止命令停止		
功能和设置	当前位置预设	(当前位置预设)指令可以用来改变当前位置到一个的值		
	Override	当允许 Override 命令在定位过程轴被执行的时候,通过应用 Override 系数将目的速度被改变。值可以被设定为 1% ~ 999%(增量为 1%)		
	数据保存	1)保存数据到中。(可以写 100000 次) 2)通过数据读取指令读到 PLC 区域 3)用支持软件读取或者保存到个人电脑的硬盘或者软盘上		

（续）

<table>
<tr><th colspan="2" rowspan="2">项</th><th colspan="3">型　　号</th></tr>
<tr><th>CJ1W-NC 113/133</th><th>CJ1W-NC 213/233</th><th>CJ1W-NC 413/433</th></tr>
<tr><td rowspan="2">外部输入/输出</td><td>输入</td><td colspan="3">为每个轴准备好下列输入：
CW 和 CCW 极限输入信号,原点接近输入信号,原点输入信号,紧急停止输入信号,定位完成输入信号,中断输入信号</td></tr>
<tr><td>输出</td><td colspan="3">为每个轴准备下列输出：
脉冲输出
CW/CCW 脉冲输出,脉冲加方向输出可以设置
偏差计数器复位或者原点调整命令输出可以根据模式选择</td></tr>
</table>

注：当执行线性内插法的时候，可以移动的距离不同。

显然，脉冲转角小，传动比大，螺距小，则脉冲当量小。脉冲当量小有利于提高定位精度。但运动总行程将缩短。因为

$$L = N \cdot P$$

这里　L——移动距离，mm；

N——脉冲总数；

P——脉冲当量。

从上式知，P 小时，同样的 L 值、N 值就要大。而 N 值是有限制的，上述特性中的位置范围即是限制。为此，在系统设计时，对 L 与 P 要合理权衡。

CJ1W 与 C200HW NC 单元性能相比，CJ1W 的性能有很大提高。

3. 操作

位控单元可进行的操作有：

（1）定位控制：定位的位移可用绝对数（即从原点到一个绝对的位置），也可用相对数（即从当前位置移到一个相对的位置）。定位控制的方法有 3 种：存储器操作、直接操作及中断操作。

1）直接操作。定位的位置和速度直接由 CPU 单元（梯形图程序）设置，并按定位操作命令执行。也有可能改变速度以及在定位正在进行过程中把移到其它位置的命令发送出去。然而，在直接操作中不可能用线性插补。图 5-80 所示为直接操作的例子。

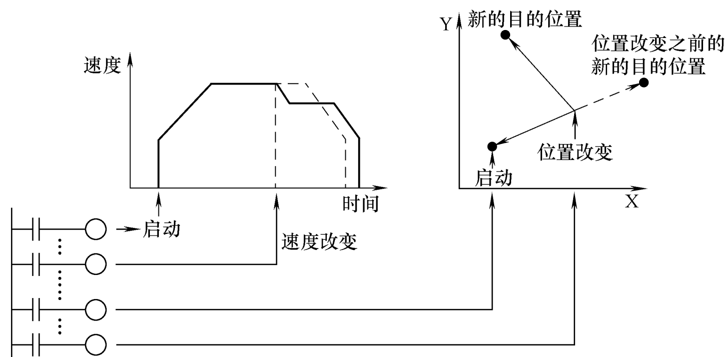


图 5-80　直接操作例子

2）存储器操作。每个定位序列（即单独的定位操作，包含位置和速度之类的数据）提前传输到 PCU，并存储。PLC 用命令调序列号，令 PCU 执行定位。序列间的转换，可做不同的设置。可选用独立定位，自动定位，或者连续定位。图 5-81 所示为存储器操作的例子。它分别调“#0”、“#1”、“#2”和“#3”定位序列。并示出不同的序列间的转换。

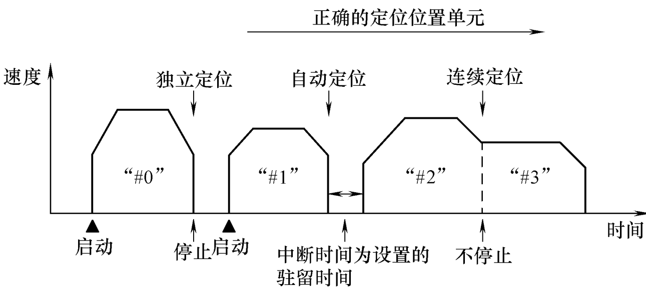


图 5-81 存储器操作例子

3）中断操作。当接收到中断输入信号时，定位继续执行一定量的脉冲后停止。如图5-82所示。

（2）速度控制。当执行一次起动的时候，脉冲以恒定的速率持续的输出。其模式依赖于对存储器操作定位序列设置的有关代码。要停止这个序列，使用 STOP（停止）指令。如图 5-83 所示。

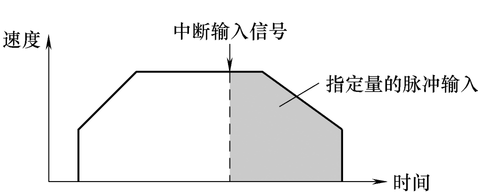


图 5-82 中断操作

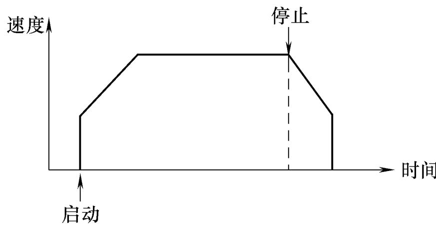


图 5-83 速度控制

（3）其它操作。有原点搜索、点动、示教、Override、预置当前位置值、间隙补偿、区域设置及减速停止。

1）原点搜索。使部件运动，找到原点停止。

2）点动。使部件以一定速度移动然后停止。

3）示教。将当前位置当作的定位序列的位置参数，如图 5-84 所示。

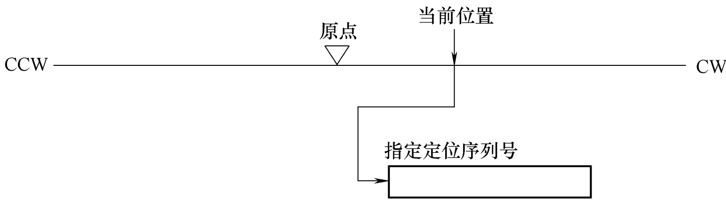


图 5-84 示教操作

4）Override：当定位过程中授权了 Override 时，目的速度被改变为 Override 速度。如图 5-85 所示。

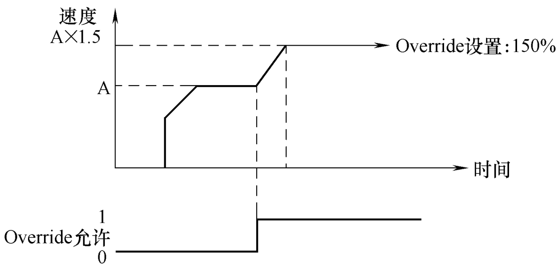


图 5-85 Override

5）预置当前位置值：改变当前位置 PRESENT POSITION PRESET（当前位置预设）命令将当前位置变到一个的位置。

6）间隙补偿：该操作用以补偿齿轮间隙，或传动装置中的其它间隙。

7）区域设置：一个区域是一个位置范围，可能定义它是为了无论何时，只要当前位置在范围之内，就将标志转为 ON。如图 5-86 所示。

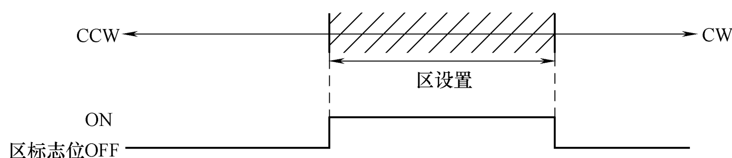


图 5-86　区域设置

8）减速停止：使定位减速到停止。如图 5-87 所示。

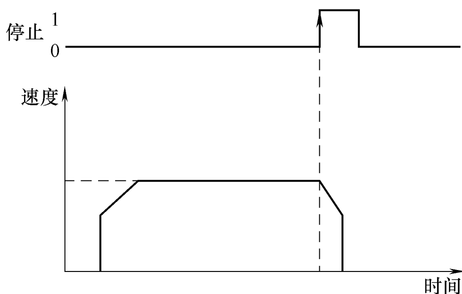


图 5-87　减速停止

4. 设定

使用 PCU，首先是单元号设定，在单元面板的单元号开关上设定。对 CJ1W-NC413/NC433，其范围为 0 ~ 94。对 CJ1W-NC113/NC133/NC213/NC233C200H 机，其范围为0 ~ 95。

单元号设定后，CJ1W-NC113/NC133/NC213/NC233 型号分配 10 个字归其使用，而 CJ1W-NC413/NC433 型号要用 20 个字。可以在上面给出的范围之内任意设定，但不能与其它单元的单元号重叠。

确定单元号，其所用的 10 个或 20 个工作通道与 100 个或 200 个数据存储区也随之确定。若单元号设为 x，则其工作通道首地址 n：$n = 2000 + x \cdot 10$

而存储区的首地址 m：

$$m = 20000 + 100x$$

这 10 或 20 个工作通道分别用作输出控制（$n \sim n+4$）及接受输入信号（$n+5 \sim n+9$）。

5. 数据

用户可以通过设置一个公共参数来选择在 CPU 单元的数据存储器区域里设置的轴参数是否传送到 PCU 中使用，或者是否使用保存在 PCU 的快闪存储器中的轴参数。图 5-88 所示 CPU 与 PCU 有关数据及其相互交换的情况。

注：用户可以通过设置一个公共参数，来选择在 CPU 单元的数据存储器区域里设置的轴参数是否传送到 PCU 中使用，或者是否使用保存在 PCU 的快闪存储器中的轴参数。

从图可知，在 CPU 单元有 4 个数据区：由单元号确定的 CIO 中的 I/O 字（操作存储器区），DM 分配给特殊 I/O 单元的区域字（公共参数区及轴参数区），用户指定的数据区域的 DM 或 EM 字及用户指定的用于传送数据的 DM 或 EM 字。

（1）操作存储器区域。特殊 I/O 单元的 I/O 区。开始字由设置的 PCU 单元号决定：

$$n = 2000 + 10 \times 单元号$$

其输出用以控制 PCU 的操作，如表 5-20 所示。其输入用以读取 PCU 的有关工作状态信息，如表 5-21 所示。在每个 I/O 刷新时，数据被更新。

DM 分配给特殊 I/O 单元的区域字包括公共参数区域及轴参数区域。

（2）公共参数区域。特殊 I/O 单元的数据存储区（DM）的 m 到 $m+3$，4 个字。用以设置与 PCU 基本操作有关的参数，见表 5-22。而 $m = D20000 + 100 \times 单元$。在使用 PCU 之前必须设置公共参数。

在设置好公共参数后，这些参数将会在下一次 PCU 被上电或者重新起动时生效。

（3）轴参数区域。特殊 I/O 单元的数据存储区（DM）的其它部分。用以设置与轴操作有关的参数，见表 5-23。

其中 I/O 设置，定义和 I/O 相关的下列项：输出脉冲选择（CW/CCW 输出，脉冲/ 方向输出）；触点类型（N. O. /N. C. ），极限输入信号，原点接近输入信号，以及原点输入信号；

图 5-88 CPU 与 PCU 数据及其相互交换

表 5-20 操作存储器区域（输出）

字							位	操 作
1 轴	2 轴		4 轴					
X 轴	X 轴	Y 轴	X 轴	Y 轴	Z 轴	U 轴		
n	n	$n+2$	n	$n+2$	$n+4$	$n+6$	00	序列号使能
							01	起动
							02	独立起动
							03	绝对移动
							04	相对移动
							05	中断进给
							06	原点搜索
							07	原点返回
							08	当前位置预置
							09	点动
							10	方向定义
							11	示教
							12	释放禁止/复位

（续）

字							位	操　　作
1 轴	2 轴		4 轴					
X 轴	X 轴	Y 轴	X 轴	Y 轴	Z 轴	U 轴		
n	n	$n+2$	n	$n+2$	$n+4$	$n+6$	13	偏差计数器复位输出/原点调整命令输出
							14	激活 Overriede
							15	停止
$n+1$	$n+1$	$n+3$	$n+1$	$n+3$	$n+5$	$n+7$	08	强迫中断
							12	写数据
							13	读数据
							14	保存数据

表 5-21　操作存储器区域（输入）

字							位	操　　作
1 轴	2 轴		4 轴					
X 轴	X 轴	Y 轴	X 轴	Y 轴	Z 轴	U 轴		
$n+2$	$n+4$	$n+7$	$n+8$	$n+11$	$n+14$	$n+17$	04	等待存储器操作标志位
							05	定位结束标志位
							06	无原点标志位
							07	原点停止标志位
							08	区域 0 监控标志位
							09	区域 1 监控标志位
							10	区域 2 监控标志位
							11	示教完成标志位
							12	标志位
							13	忙标志位
							14	数据传送标志
							15	减速停止执行标志
$n+3$	$n+5$	$n+8$	$n+9$	$n+12$	$n+15$	$n+18$	08	CW 极限输入信号
							09	CCW 极限输入信号
							10	原点接近输入信号
							11	原点输入信号
							12	中断输入信号
							13	紧急停止输入信号
							14	定位完成信号
							15	偏差计数器复位输出/原点调整命令输出
$n+4$	$n+6$	$n+9$	$n+10$	$n+13$	$n+16$	$n+19$	00 到 15	代码

表 5-22 公共参数区域

字(所有模式均相同)	名　称	配置/解释
m	操作数据区域定义	定义操作数据被设置的存储器区域。从以下所示中选择一项。 0000:分配给特殊 I/O 单元的数据区域字(固定) 000D:用户定义的数据存储器区域字 0X0E:用户定义的 EM 区域字
$m+1$	操作数据区域开始字	定义操作数据区域的开始字,如果 $m=000D$ 或者 0X0E 以十六进制的形式定义操作数据区域的区域开始字
$m+2$	轴参数定义	定义用作轴参数的数据的位置。从以下所示中选择一项 • 在 PCU 的快闪存储器中保存的轴参数数据 • 在 CPU 单元的数据存储器区域设置的轴参数数据 • PCU 的默认设置
$m+3$	未被使用	这个区域未被使用。被设置为 0000

表 5-23 轴参数区域

字(PCU 中的地址)				名　称	数据大小
X 轴	Y 轴	Z 轴	U 轴		
$m+4$(0004)	$m+32$(0020)	$m+60$(003C)	$m+88$(0058)	I/O 设置	一个字
$m+5$(0005)	$m+33$(0021)	$m+61$(003D)	$m+89$(0059)	操作模式选择	一个字
$m+6$(0006)	$m+34$(0022)	$m+62$(003E)	$m+90$(005A)	最大速度	两个字
$m+8$(0008)	$m+36$(0024)	$m+64$(0040)	$m+92$(005C)	初始速度	两个字
$m+10$(000A)	$m+38$(0026)	$m+66$(0042)	$m+94$(005E)	原点搜索高速度	两个字
$m+12$(000C)	$m+40$(0028)	$m+68$(0044)	$m+96$(0060)	原点搜索接近速度	两个字
$m+14$(000E)	$m+42$(002A)	$m+70$(0046)	$m+98$(0062)	原点补偿	两个字
$m+16$(0010)	$m+44$(002C)	$m+72$(0048)	$m+100$(0064)	间隙补偿	一个字
$m+17$(0011)	$m+45$(002D)	$m+73$(0049)	$m+101$(0065)	间隙补偿速度	两个字
$m+19$(0013)	$m+47$(002F)	$m+75$(004B)	$m+103$(0067)	加速/减速曲线	一个字
$m+20$(0014)	$m+48$(0030)	$m+76$(004C)	$m+104$(0068)	原点搜索加速时间	两个字
$m+22$(0016)	$m+50$(0032)	$m+78$(004E)	$m+106$(006A)	原点搜索减速时间	两个字
$m+24$(0018)	$m+52$(0034)	$m+80$(0050)	$m+108$(006C)	位置监控时间	一个字
$m+25$(0019)	$m+53$(0035)	$m+81$(0051)	$m+109$(006D)	CCW 软件限位	两个字
$m+27$(001B)	$m+55$(0037)	$m+83$(0053)	$m+111$(006F)	CW 软件限位	两个字
$m+31$(001F)	$m+59$(003B)	$m+87$(0057)	$m+115$(0073)	初始脉冲定义	一个字

当输入紧急停止信号时，偏差计数器输出控制；当输入紧急停止信号或者极限信号时原点丢失。

其中模式设置，用以设置电动机驱动器的选用以及定义原点检测方法。有 4 种模式：当使用步进电动机驱动器时，设置为模式 0；当使用伺服驱动器，并且不使用定位完成信号将驱动器输入和偏差计数器复

位输出信号连接起来的时候，设置为模式 1；当使用伺服驱动器，并且将驱动器输入和偏差计数器复位输出信号连接起来的时候，设置为模式 2；当使用伺服驱动器，并且同时使用原点调整命令时设置为模式 3。默认设置为模式 0。

（4）操作数据区域。在公共参数区域中，指定的 DM 或 EM 字。分公共部分与个别部分。公共部分用以设置数据传送有关操作，如表 5-24 所示。个别部分用以设置位置，速度以及加速/ 减速时间，以及存储器操作的位置号，如表 5-25 所示。同时包含 PCU 状态数据，例如目前的位置和那些目前执行的位置序列号。在每个 I/O 刷新时，数据被更新。设置数据被起动并且在每个运行的起动时使用。

（5）用户指定的用于传送数据的 DM 或 EM 字。操作数据区域（公共部分）指定的 DM 或 EM 区，用以暂存有关存储操作的有关数据。以便用操作存储器区域的输出命令，在实施存储操作前传送给 PCU 的内部存储器操作区域（可以被存储到 PCU 的快闪存储器中）。

存储器操作区域的内部地址，如表 5-26 所示。包含 6 种类型的数据：定位序列、速度、位置、加速时间、减速时间及驻留时间。各含 100 组。

表 5-24　操作数据区域（公共部分）

字		操　　作
l	写字数	定义从 CPU 单元写到 PCU 中的字数
$l+1$	写源区域	定义了从 CPU 单元写到 PCU 中的数据区域
$l+2$	写源字	定义了从 CPU 单元写到 PCU 中数据区域开始字
$l+3$	写目标地址	定义了写入 PCU 中的数据的地址
$l+4$	读字数	定义从 PCU 读入 CPU 单元的字数
$l+5$	读源区域	定义从 PCU 中读数据的地址
$l+6$	读目示区域	定义从 PCU 中读数据时输出数据的区域
$l+7$	读目标字	定义从 PCU 中读时用来输出数据的字

表 5-25　操作数据区域（个别部分）

字							名　　称	数据大小
1 轴	2 轴		4 轴					
X 轴	X 轴	Y 轴	X 轴	Y 轴	Z 轴	U 轴		
$l+8$	$l+8$	$l+20$	$l+8$	$l+20$	$l+32$	$l+44$	位置	两个字
$l+10$	$l+10$	$l+22$	$l+10$	$l+22$	$l+34$	$l+46$	速度	两个字
$l+12$	$l+12$	$l+24$	$l+12$	$l+24$	$l+36$	$l+48$	加速时间	两个字
$l+14$	$l+14$	$l+26$	$l+14$	$l+26$	$l+38$	$l+50$	减速时间	两个字
$l+16$	$l+16$	$l+28$	$l+16$	$l+28$	$l+40$	$l+52$	存储操作起动序列号	一个字
$l+17$	$l+17$	$l+29$	$l+17$	$l+29$	$l+41$	$l+53$	Override 速率	一个字
$l+18$	$l+18$	$l+30$	$l+18$	$l+30$	$l+42$	$l+54$	示教地址	一个字
$l+19$	$l+19$	$l+31$	$l+19$	$l+31$	$l+43$	$l+55$	未被使用	一个字
$l+20$	$l+32$	$l+36$	$l+56$	$l+60$	$l+64$	$l+68$	当前位置	两个字
$l+22$	$l+34$	$l+38$	$l+58$	$l+62$	$l+66$	$l+70$	位置序列	一个字
$l+23$	$l+35$	$l+39$	$l+59$	$l+63$	$l+67$	$l+71$	输出代码	一个字

表 5-26　存储器操作数据

PCU 内部地址				名　　称	数据大小
X 轴	Y 轴	Z 轴	U 轴		
1000	2000	3000	40000	定位序列号 0 ~ 99	三个字
112C	212C	312C	412C	速度号 0 ~ 99	两个字
11F4	21F4	31F4	41F4	位置号 0 ~ 99	两个字
12BE	22BE	32BE	42BE	加速时间号 1 ~ 9	两个字
12D2	22D2	32D2	42D2	减速时间号 1 ~ 9	两个字
12E5	22E5	32E5	42E5	驻留时间号 1 ~ 19	一个字

其中定位序列 3 个字的含义，以 X 轴的序列 1 为例，如下：

	15　　12	11　　08	07　　04	03　　00
1000	轴定义	输出代码	位置指定	完成代码
1001	驻留时间号		加速时间号	减速时间号
1002	初始速度号		目标速度号	

具体各项的内容为：

轴定义：　将活动轴设置为“1”

位 15：U 轴；14：Z 轴；13：Y 轴；12：X 轴

输出代码：　00 ~ 0F 十六进制

位置指定：　指定每个轴的位置数据是绝对或者是相对位置。

位 7：U 轴；6：Z 轴；5：Y 轴；4：X 轴

0：绝对位置；1：相对位置

完成代码：　00 ~ 06Hex（0 ~ 6）

驻留时间号　00 ~ 13Hex（0 ~ 19）

加速时间号　0 ~ 9Hex（0 ~ 9）

减速时间号　00 ~ 09Hex（0 ~ 09）

初始/目标速度号：00 ~ 63Hex（0 ~ 99）

速度的设定如下：

15　　00	15　　00
速度#0（最左边的字）	速度#1（最右边的字）

设置速度#0（以 pps 为单位）该速度可以用两个字 32 位无符号十六进制数设置在 1 ~ 1000000p/s。设置范围：00000001 ~ 000F4240 十六进制（1 ~ 1000000）

位置设定如下：

15　　00	15　　00
位置#0（最左边的字）	位置#1（最右边的字）

设置位置#0（以脉冲为单位）位置可以用两个字，32 位有符号十六进制数据设置在范围 − 1073741823 ~ 1073741823 脉冲之间。

设置范围：C0000001 ~ 3FFFFFFF（ − 1073741823 ~ 1073741823）

前已提及，位控单元的有多种操作，现仅对直接操作及存储操作讨论如下：

1）直接操作。直接操作进行位置的方法是：先在操作数据区域设定位置、速度、加速/减速时间等数据，通过传送指令及 I/O 刷新输出，把操作数据传给 PCU；后使用操作存储区起动操作，具体实现定位过程。图 5-89 所示为这两步骤的过程。

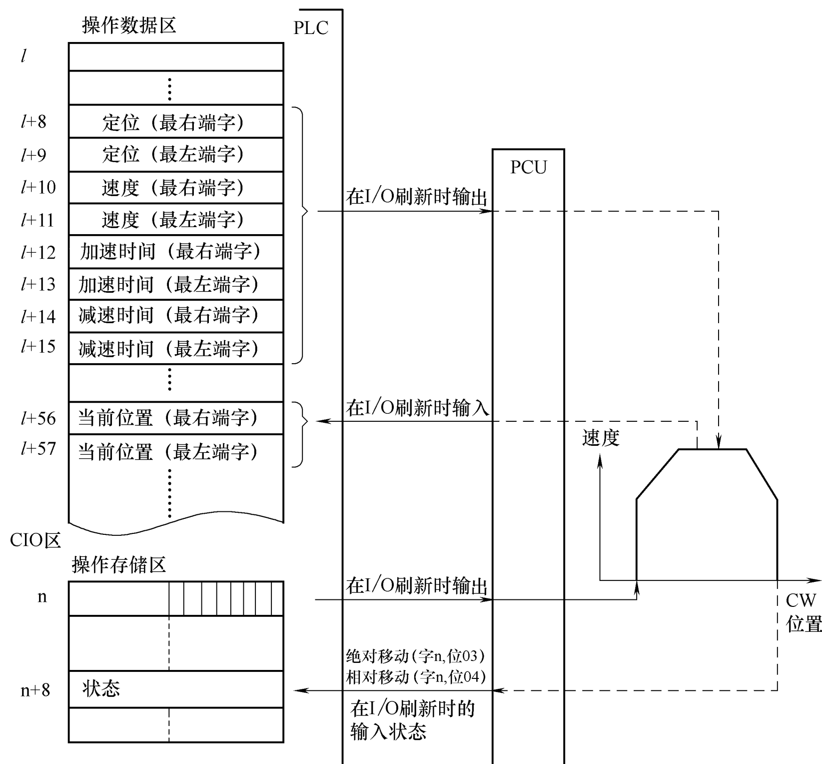


图 5-89　直接操作过程

直接操作实现步骤：

（a）设置公共参数：

m：设置操作数据区到 DM 或者 EM。

$m+1$：设置操作数据区（l）开始字。

$m+2$：指定轴参数。

（b）重新上电或者重新起动。

在上面被设置在（l）中的公共参数区中的数据可用。

（c）设置操作数据区（以单轴为例）：

在 $l+8$ 和 $l+9$ 中，设置位置。

在 $l+10$ 和 $l+11$ 中，设置速度。

在 $l+12$ 和 $l+13$ 中，设置加速时间。

在 $l+15$ 和 $l+16$ 中，设置减速时间。

（d）执行绝对移动或相对移动。

把绝对移动命令位（字 n，位 03）或相对移动命令位（字 n，位 04）置位。

直接操作还有一些细节，如多重起动、加减速时间计算，可参阅有关说明。

2）直接操作例子。使用的位置控制单元是一个 NC113，轴参数设置用缺省值。X 轴以 15000p/s 的速度，相对移动 135000 脉冲，如图 5-90 所示。

单元号及有关设定如表 5-27 所示。位置、速度等设定如表 5-28 所示。梯形图程序如图 5-91 所示。从图知，这个程序是很简单的。主要的工作是做好设定及设计好参数。

3）存储操作

存储操作，应先把与定位序列有关的位置和速度等定位数据，预先传送到位置控制单元的内部存储器中，然后，根据操作存储区的控制命令，起动定位序列实施定位。其过程如图 5-92 所示。

存储操作存储的数据量大，最多可进行 100 组操作，而不像直接操作，每个起动命令只能进行一个操作。

图 5-90　定位图

表 5-27　单元号及有关设定

项　　目	细　　节
位置控制单元的单元号	3 公共参数区：D20300 ~ D20302 操作存储区：CIO 2030 ~ CIO 2034 以上区域由设置单元号自动设置
操作数据区指定	D20300 | 0 | 0 | 0 | D | – – – DM区 D20301 | 0 | 1 | F | 4 | – – – D00500
轴参数指定	D20302 | 0 | 0 | 0 | 0 | – – – 使用位置控制单元默认参数

表 5-28　位置、速度等设定

名　　称	结　　构	数据管理	内容
位置	最左端 15 l+9 00　最右端 15 l+8 00 设置范围： C0000001～3FFFFFFF Hex (−107374182～1073741823脉冲)	00508 00509	0F58 0002
速度	最左端 15 l+11 00　最右端 15 l+10 00 设置范围： 1～7A120 Hex (1～500000 p/s)	00510 00511	3A98 0000
加速时间	最左端 15 l+13 00　最右端 15 l+12 00 设置范围： 0～3D090 Hex (0～250000 ms)	00512 00513	0064 0000
减速时间	最左端 15 l+15 00　最右端 15 l+14 00 设置范围： 0～3D090 Hex (0～250000 ms)	00514 00515	0064 0000

4）存储操作例子

本例为使用 NC113 位置控制单元实现如图 5-93 所示的位置控制。

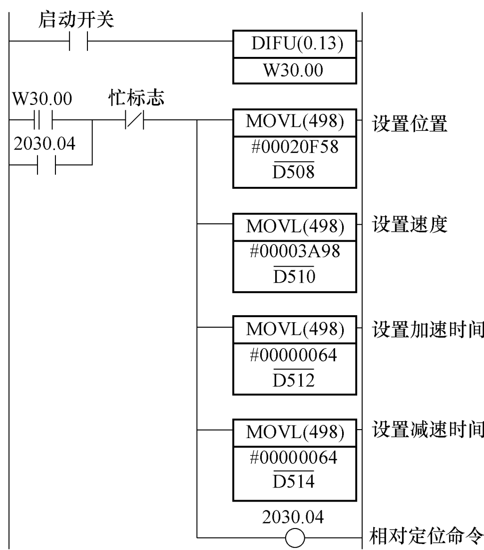


图 5-91　直接操作程序

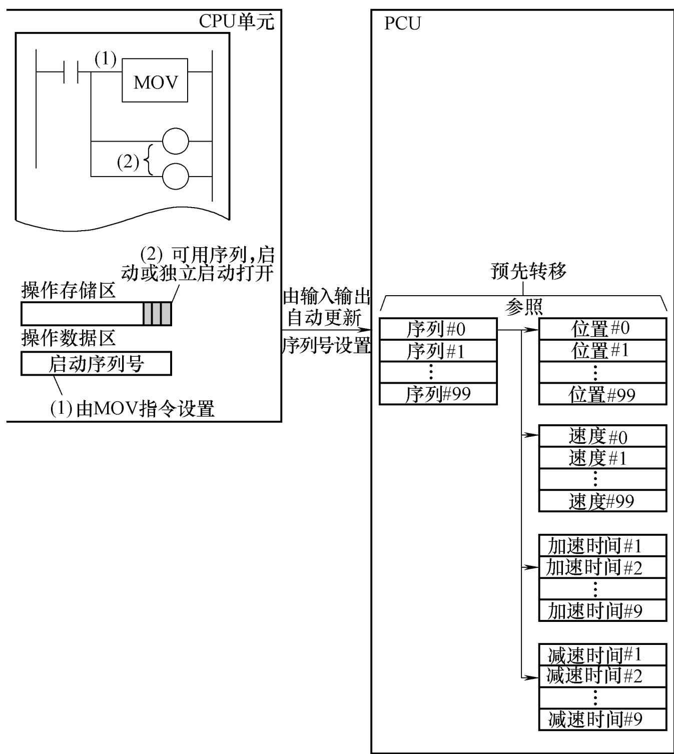


图 5-92　存储操作过程

定位序列设置见表 5-29。这里设置了序列 10 及 11。单元号及有关设置见表 5-30。定位序列 10、11 的细节设置如表 5-31 所示。速度设置如表 5-32 所示。

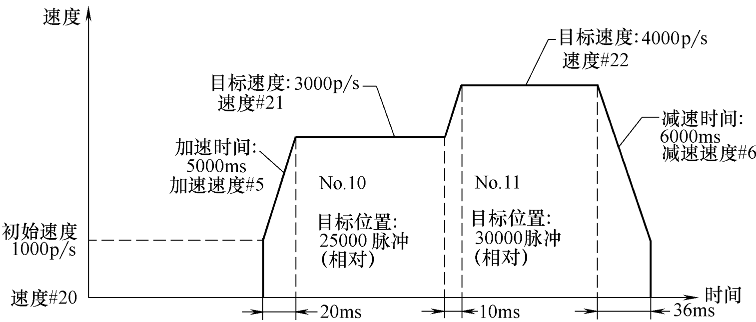


图 5-93　位置控制要求

表 5-29　定位序列设置

数　　据	设定值	
	序列#10	序列#11
轴定义	1(X 轴)	1(X 轴)
输出码	0	0
位置定义	1(X 轴,相对位置)	1(X 轴,相对位置)
完成码	2(连续)	3(块结束)
驻留时间号	00	00
加速时间号	5	5
减速时间号	6	6
初始速度号	14	14
目标速度号	15	16

表 5-30　单元号及有关设置

<table>
<tr><th colspan="2">项　　目</th><th>细　　节</th></tr>
<tr><td colspan="2">位置控制单元的单元号</td><td>单位 1:普通参数区:D20100 ~ D20102
操作存储区:CIO 2010 ~ CIO 2014(字)
(根据单元号设置上面区域被自动分配)</td></tr>
<tr><td rowspan="3">普通参数</td><td>操作数据区定义</td><td>D20100 [0 | 0 | 0 | D] DM 区</td></tr>
<tr><td>操作数据区的开始字</td><td>D20101 [0 | 0 | C | 8] D00200</td></tr>
<tr><td>轴参数字义</td><td>D20102 [0 | E | 0 | 0] 使用保存在PCU中的参数
(用于X ,Y, Z轴的默认设定值)</td></tr>
</table>

表 5-31　序列设置细节

数据	数据构造	值设定	地址
序列号 10	15 12 11 08 07 04 03 00 轴指定 | 输出码 | 位置指定 | 完成码 驻留时间号 | 加速时间号 | 减速时间号 初始速度号 | 目标速度号	1012 0056 1415	101E 101F 1020
序列号 11		1013 0056 1416	1021 1022 1023

数据传送梯形图程序如图 5-94 所示。而起动定位梯形图程序如图 5-95 所示。程序也都比较简单。

表 5-32　速度设置

数据	数据构造	设定值（脉冲/秒）	值设定	地址
速度数据号 20	15 最左端 00 15 最右端 00 速度数据 | (p/s) 无符号32位二进制数据 设置范围 1～7A120 Hex (1～500000p/s)	1000	03E8 0000	1154 1155
速度数据号 21		3000	0BB8 0000	1156 1157
速度数据号 22		4000	0FA0 0000	1158 1159

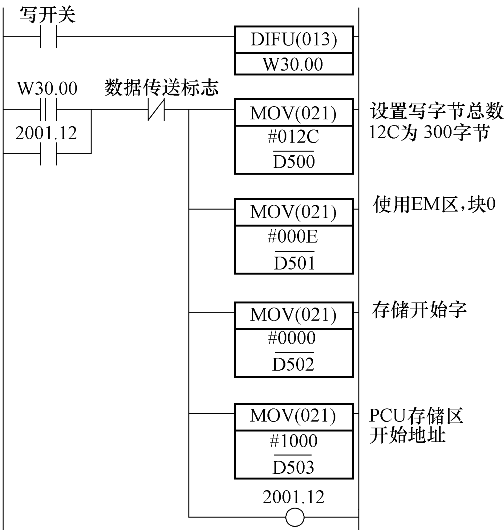


图 5-94　数据传送程序

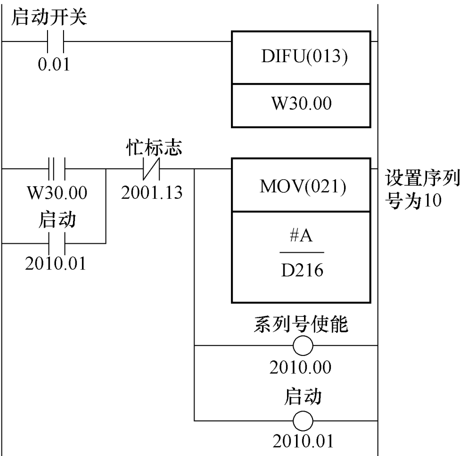


图 5-95　起动定位程序

5. 6. 2　运动控制单元

OMRON 的运动控制单元称 MC，型号也很多，有双轴的（如 CS1W-MC221），也有 4 轴的（CS1W-MC421）。它处理的虽为脉冲量，但实际输出是模拟量。不能用步进电动机，只能用伺服电动机。因为后者自带有旋转编码器，可用脉冲的形式反馈角位移信号。其定位程序主要是用 G 语言编程，并要用专门的编程软件。它除了作直线插补外，还可作圆弧插补，空间螺旋线插补。可用于要求较高的运动控制。

1. 基本原理

MC 用的是半闭环控制。图 5-96 所示为它的原理。

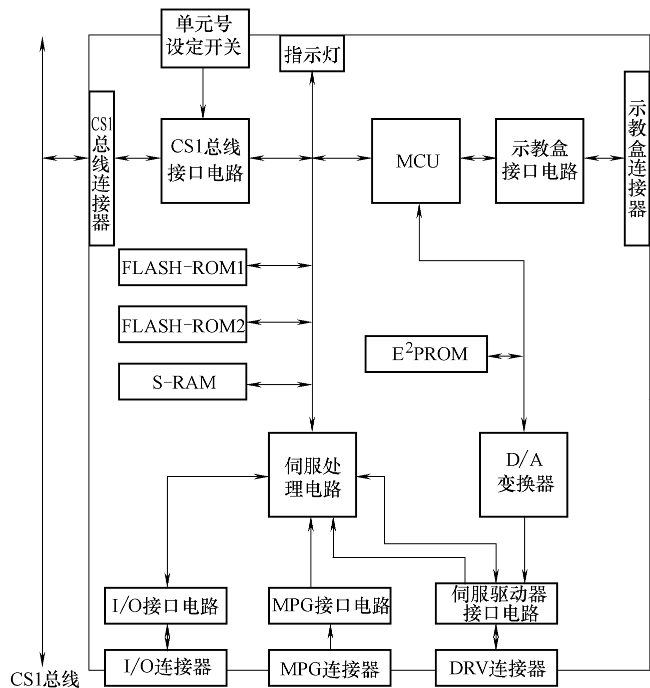


图 5-96　MC 工作原理

MPG—手工脉冲生成器　MCU—系统控制用微处理器　FLASH-ROM1—存储系统程序　FLASH-ROM2—存储 G 语言程序系统参数位置数据　S-RAM—临时存储　E^2PROM—出错记载

从图知，MC 有微处理器、内存、处理电路、变换器及很多接口。可通过 CS1 总线与 PLC 连接，从而，可与 PLC 的 CPU 单元交换数据。自身的内存可存储 G 语言程序、系统参数及运行数据。自身的 MCU 可运行程序，管理系统。伺服处理电路，可处理普通开关量信号、脉冲信号。D/A 变换器，可把数字转换为模拟信号，并可通过 DRV 接口加载给伺服电动机驱动器。

另外，还有 MPG 连接器。用它可连接手工脉冲生成器，以使系统可用手工生成的脉冲进行调试。还有示教盒连接器。用它可连接示教盒，以用示教盒对系统进行控制。

系统工作时，MCU 可运行 G 语言编的程序或执行 CPU 来的命令，以实现控制。实现控制的方法是把输出，经 D/A 变换，输出给伺服驱动器。伺服驱动器驱动伺服电动机转动。伺服电动机转动后，与其同步转动的旋转编码器产生脉冲。此脉冲经 DRV 接口、伺服处理电路，反馈给 MPU。直到所确定的输出与反馈脉冲对应相当时，D/A 的输出为 0，即完成一个运动位置的控制。图 5-97 所示为 MC 实现控制的简单过程。

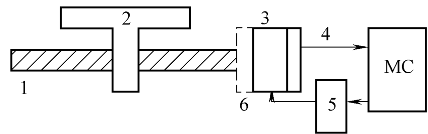


图 5-97　MC 实现控制过程

1—滚珠丝杆　2—工作台　3—伺服电动机　4—旋转编码器反馈脉冲　5—伺服驱动器　6—减速器

图 5-98 示的为 MC 实现控制过程中信号处理关系。

从图知，当给出目标位置脉冲后。由于电动机未转动，无反馈脉冲，故误差计数器出现误差脉冲。经 A/D 转换，产生速度控制电压，进而使电动机转动。转动后的反馈脉冲将消除误差。误差没有，速度控制电压为 0，电动机转动也停止。可知，给出目标脉冲数，必然会使伺服电动机转动，直到反馈的与给定的相等为止。

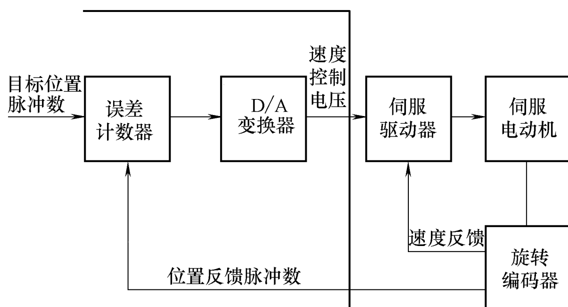


图 5-98　MC 控制过程信号处理关系

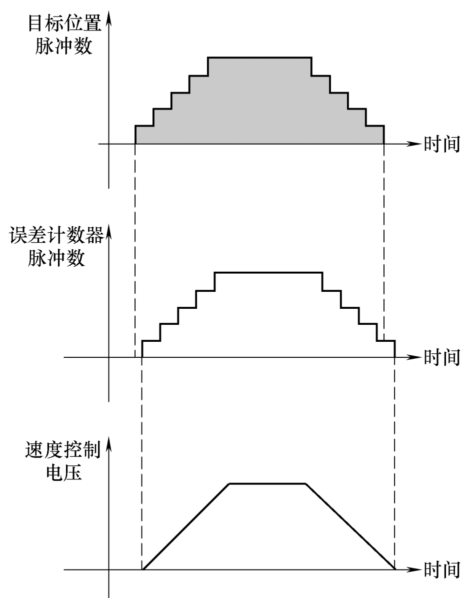


图 5-99　目标脉冲数、误差脉冲数及电压间关系

但如果给出的目标脉冲加速增加，则误差计数器误差脉冲也将增加，进而速度控制电压、电动机转速将增大。如果目标脉冲等速给出，或减速给出，则误差计数器误差脉冲将不变，或减小，进而速度控制电压、电动机转速不变，或减小。图 5-99 所示为这些目标位置脉冲数、误差计数器脉冲数及速度控制电压间的关系。

2. 系统配置

图 5-100 所示为系统可能的配置。

3. 主要特点（指 CS1 的 MC）

（1）使用多任务 G 语言编程，而不用梯形图编程，这既简化编程，又可减少 CPU 在执行梯形图程序的负担。

（2）一个定位命令输出后，下一个定位操作命令即可起动，而不必等待这个定位完成。因而可实现高速的顺序操作。

（3）支持绝对式编码器，也支持增量式编码器。

（4）响应 CPU 单元的指令的时间很短，双轴的在 8ms，4 轴的在 12ms 即可响应。

（5）反馈编码器脉冲频率最大可达 500kp/s。

（6）当定位完成或通过一个特定区域，MC 向 CPU 输出 D 码，可使 CPU 单元产生中断。所以，用此方法可使 MC 与 CPU 实现理想的同步。

（7）可与 CXP 同时使用 MC 的 CX-Motion 编程及监控软件。可追踪 500 项数据，为伺服系统调试提供方便。CX-Motion 可存储程序与定位数据，一旦 MC 需要，可自动下载给 MC。

（8）当部件运动时，可使用示教盒自动建立位置数据。

（9）可用手动脉冲发生器定位及同步操作。

4. 主要操作

MC 的主要可进行的操作有：

（1）点到点（Point To Point）操作。PTP 控制，每个轴分别独立控制，只控制运动的速度及起点和终点位置。如从原点出发，都以相同速度 50，X 移动 100，Y 移动 50，就是 PTP。图 5-101 所示即为这个运动的情况。

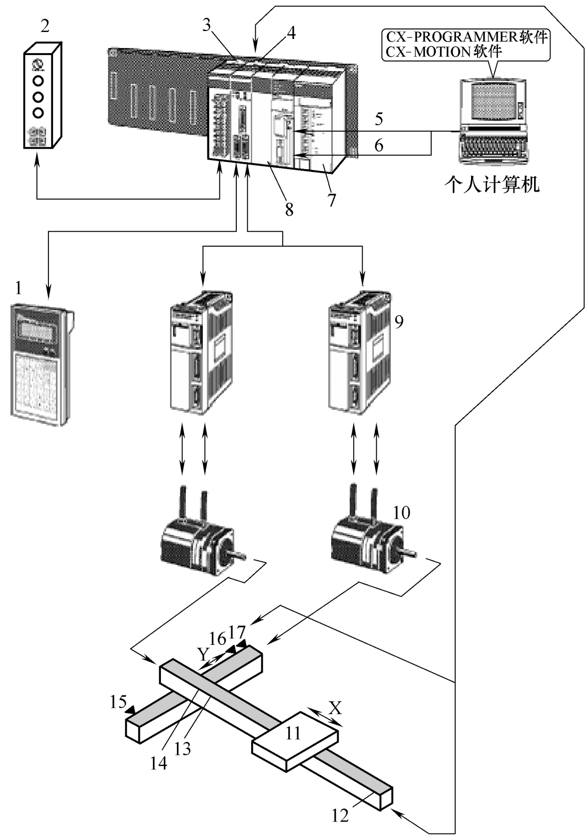


图 5-100　MC 系统配置

1—示教盒　2—控制盒　3—输入单元　4—MC 单元　5—外设口　6—RS232 口　7—PLC 电源　8—CPU 单元　9—伺服驱动器　10—伺服电动机　11—工作台　12—X 轴原点开关　13—X 轴顺时针方向极限行程开关　14—X 轴逆时针方向极限行程开关　15—Y 轴顺时针方向极限行程开关　16—Y 轴原点行程开关　17—Y 轴逆时针方向极限行程开关

（2）连续路径（Continuous Path，CP）操作。这种操作不仅控制运动的起点、终点，还要控制运动轨迹。为此，可进行直线插补、圆弧插补，如图 5-102 所示。

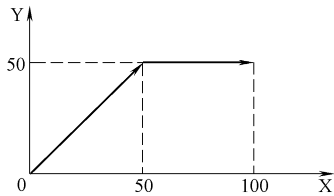


图 5-101　PTP

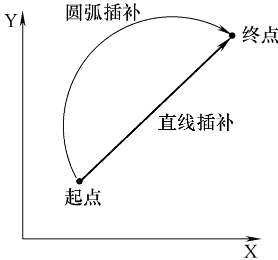


图 5-102　直线、圆弧插补

（3）纺线操作。图 5-103 所示为纺线控制。这里，纺线轴定速转动，而移动架等速左右直线运动。两者配合，可实现纺线功能。

（4）多圈螺旋线插补操作。图 5-104 所示即为此操作。用于对绕线机的控制。

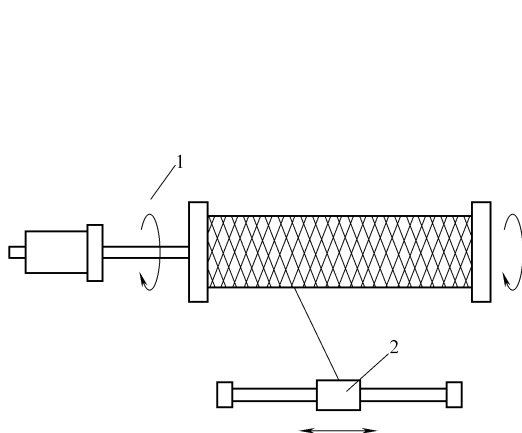


图 5-103　纺线控制

1—纺线轴　2—移动架

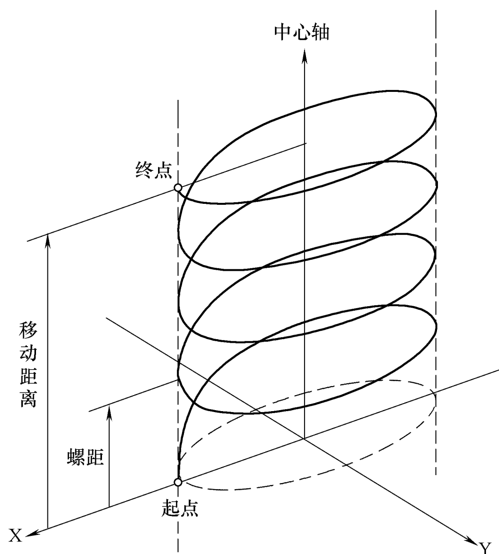


图 5-104　多圈螺旋线插补

此外，还有原点查找、点动、错误计数复位、现位置预设等操作。

MC 的操作模式有手动及自动两种。手工执行为手工模式，执行 CPU 单元指令或示教盒指令；自动执行为自动模式，执行预编的 G 语言程序。图 5-105 所示为这两种模式下操作示意。

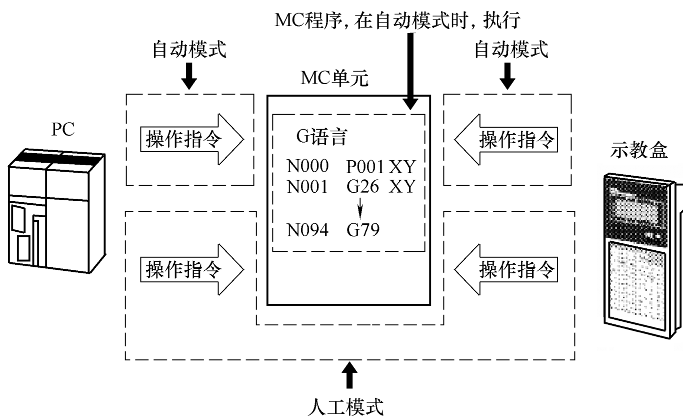


图 5-105　两种模式下操作

手工模式下的操作有：

点动，靠人工操作连续移动；处理进给，用 MPG 移动；减速停止，根据命令减速到停止；人工查找原点，绝对旋转编码器，增量旋转编码器均可查找；等 11 种操作。

自动模式下的操作有：线性插补定位，可在 2、3 及 4 轴间实现线性插补定位，进给率可指定；圆弧插补定位，可进行 2 轴顺时针或反时针插补，进给率也可指定；螺旋插补定位，可进行 2 轴顺时针或反时针及 1 轴线性插补（仅 CS1W-MC421），进给率也可指定；纺线功能，可执行操作纺线操作；速度控制，可控制最大移动速度；等等。

5. 主要特性及内存区

CS1W-MC221 及 CS1W-MC421 的主要特性见表 5-33。

表 5-33 CS1W-MC221、CS1W-MC421 主要特性

项目		特性	
		CS1W-MC221	CS1W-MC421
自动操作模式(对各轴)		有 11 种自动模式指令,查找原点等 自动循环从检测到 CPU 单元起始位或示教盒命令开始	
编码器接口		最大响应脉冲频率:500kp/s 脉冲比率:1,2 或 4	
控制单位	最小设定单位	1,0.1,0.01,0.001,0.0001	
	单位	mm,inch,degree,pulse	
最大命令值		-39999999 ~ +39999999(当最小单位为 1 时)	
控制轴数		最多 2 轴	最多 4 轴
速度		1 ~2000000p/s(当比率为 4 时)	
加、减速度		梯形或 S 形	
加、减速度时间		0 ~100000ms(2-ms 增量)	
任务程序管理	任务数	最多 2	最多 4
	程序数	当 1 个任务时,100 当 2 个任务时,50	当 1 个任务时,100 当 2 个任务时,50 当 3 个任务时,33 当 4 个任务时,25
	程序容量	当 1 个任务时 2000 块 当 2 个任务时 1000 块 在单程序中,最多 800 块	当 1 个任务时,2000 块 当 2 个任务时,1000 块 当 3 个任务时,666 块 当 4 个任务时,500 块 在单程序中,最多块 800
	定位数据能力	总共 2000 位置	
	寄存器数	32(主要用于指定定位数据)	
	子程序层数	最多 5 层	

（1）MC 内存区比 PCU 大的多。要存储 4 种数据，分别是系统参数：含单元参数、内存管理参数、轴参数。轴参数含有 MC 要使用的系统信息，如使用轴数、任务数、进给率及操作范围。在 MC 的内部地址为 4000 ~4684。定位数据：可存 2000 个位置地址。用 G 语言时，用 A0000 ~ A1999 表示。在 MC 的内部地址为 0000 ~1999。监控数据：是只读数据。含有 MC 单元不同类型的数据。诸如 I/O 监控数据，任务及 MC 出错时的轴出错码。在 MC 的内部地址为 6000 ~6081。命令数据：用以传送位置数据写系统参数到闪烁内存，用 IOWR 指令写特定命令，执行自动装载及其它功能。在 MC 的内部地址为 6100 ~6120。

（2）数据传送方法。联机使用 CX-Motion 软件下载、上载数据；使用 PLC 执行智能读写，即 IOWR/IORD 指令传送数据；使用 PLC 执行智能写，即 IOWR 指令，写指令区（#17D6）传送给定位数据。

图 5-106 所示为执行 IOWR 指令传送定位数据的示意。传送其它数据，或读数据也类似。

这里 C——定位数据的在 MC 中的首地址

S——含有定位数据的 PLC DM 区的首地址

D——占两个字，D 指定目标单元号，D+1 指定传送数据总数（以字为单位）

目标单元号就是设定的 MC 单元号。因为一个 PLC 可安装多个 MC 单元，故须指定这个单元号，以示

区别。定位数据总数，每项 3 个字计。

（3）任务配置。在 MC 使用前须先配置任务。对 CS1W-MC421，最多可设置 4 个任务。任务是执行程序的单位。如有 4 个任务同步执行，MC 单元将具有同时控制 X、Y、Z 及 U4 轴的功能。而同一轴只能设定一个任务。CS1W-MC221 最多可设 2 个任务，仅能同时控制 X 与 Y 轴。

（4）接口区。需要时，PLC 可通过接口区经 I/O 刷新控制 MC。它的分布由 MC 单元号的设定决定。CS1W-MC421 分配 50 个字，而 CS1W-MC221 分配 30 个字。单元号与接口区开始字地址 n 的关系是：

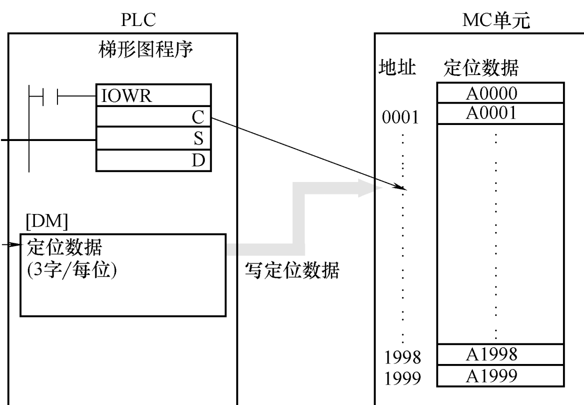


图 5-106　执行 IOWR 指令传送定位数据示意

$$n = \text{CIO}2000 + 10 \times \text{单元号}$$

其中：

对 CS1W-MC421，$n \sim n+17$ 为输出区；$n+18 \sim n+47$ 为输入区而 $n+48 \sim n+49$ 系统留用。

对 CS1W-MC221，$n \sim n+9$ 为输出区；$n+10 \sim n+25$ 为输入区而 $n+26 \sim n+29$ 系统留用。

CS1W-MC221 单元号要占 3 个字，而 CS1W-MC421 要占 5 个字。故 CS1W-MC221 的最大单元号只能是 93，指定后，94 和 95 不能作为它用。而 CS1W-MC421 的最大单元号只能是 91，指定后，92、93、94 和 95 不能作为它用。还要所有单元号不能重复。

6. 使用

使用 MC 单元大体过程是：模块安装，机号设定，接线，接通电源，建立 I/O 表，确定任务号，利用 CX_MOTION 软件设置系统参数并下载给 MC，利用 CX_MOTION 软件用 G 语言编程并下载给 MC，利用CX_PROGRAMMER 软件编梯形图程序下载给 PLC，监控、调试程序。

G 语言是 NC 的语言，在运动控制中得到广泛的运用。它的特点是编程容易，只要先输入“G”，后加两个数字功能码，再加上所需要的参数，就是一条指令。MC 单元用的 G 语言的功能码是，从 G00 ~ G91 。如 PTP，其功能码为 G00。

例：用 G 语言编的一小段程序。具体如下。在此例子程序中，已对各个字段的含义作了说明。至于各个语句的功能如表 5-34 所示。

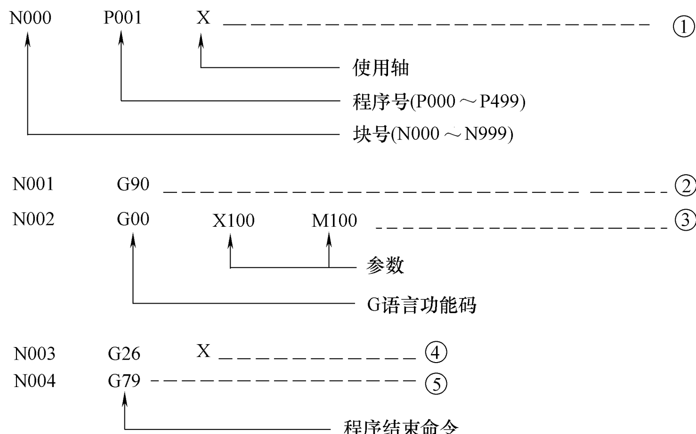


表 5-34　例 1 程序各语句功能

	MC 程序块	功　　能
①	N000 P001 X	申明程序号为 N000 及使用 X 轴
②	N001 G90	指定用绝对坐标定位
③	N002 G00 X100 M100	在 X 轴上,移动到坐标 100,当定位完成时,输出 M 码 100,执行下一块用来自 PLC 的命令 M 码复位
④	N003 G26 X	返回到原点
⑤	N004 G79	程序结束

图 5-107 所示为执行例 1 程序在 X 轴上所实现的运动。

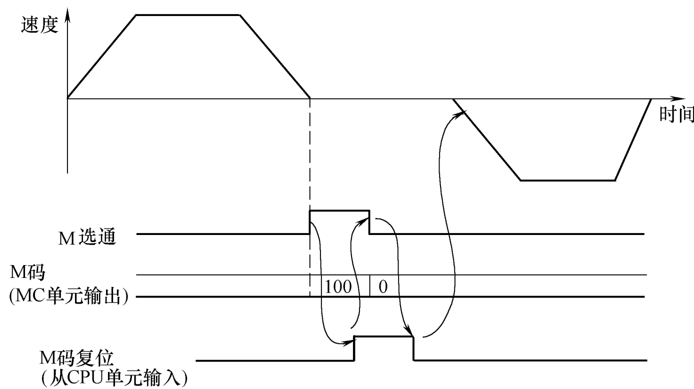


图 5-107　执行例 1 程序在 X 轴上实现的运动

以下以两轴系统使用 CS1WMC221 为例，看 MC 单元是怎么使用的。MC 执行如下操作：

（1）每轴回到原点（0，0）；

（2）X，Y 移动到（400，200）钻孔；

（3）移动下一位置（100，200）；

（4）移动到最后一个位置（200，400）；

（5）回原点。

为此，模块安装、机号设定、接线、接通电源、建立 I/O 表等先要进行。然后用 CX_MOTION 软件建立工程，并作有关设定。

运行 CX_MOTION 软件，建立工程，选 PLC，选 MC，工程如图 5-108 所示。

这时用鼠标左键双击“参数（Parameter Set1）”，将出现图 5-109 所示参数设定画面。

在其上可对使用轴数、任务数、任务内存范围等进行设定。并传送给 MC。之后是编写、编译及存储 MC 程序。登记及存储工程。

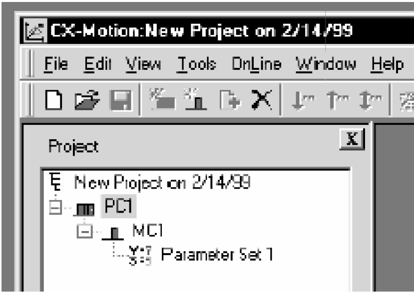


图 5-108　新工程画面

编写程序也是使用软件，打开 G 程序编辑窗口进行。本例的 G 语言程序如下：

```
N000 P001 XY // 声明程序号为 P001,用于 X、Y 轴控制
  *001 例子程序// 这是注解
  N002 G04 5   //   等待 5 秒
  N003 G26 XY   // 回 X、Y 轴各自原点
```

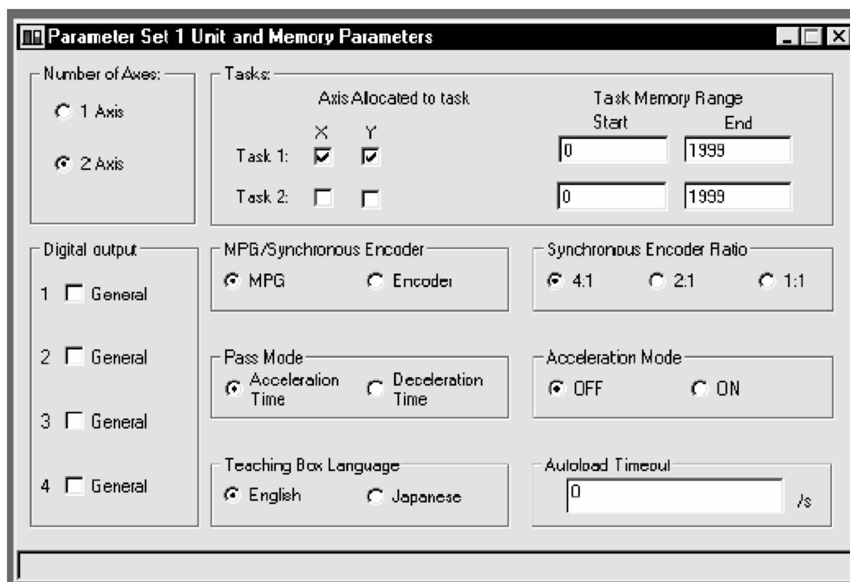


图 5-109　参数设定画面

N004 G11 // 为定位选择停止模式

N005 G01 X400 Y300 F30 // 用速度 30 移动到 X400,Y300

N006 G01 X100 Y200 F10 //用速度 10 移动到 X100,Y200

N007 G01 X200 Y400 F30//用速度 30 移动到 X200,Y400

N008 G026 XY // 回 X、Y 轴各自原点

N009 G79 //结束程序

这里每条指令//后为便于理解所加的注解。G 语言程序不能加注解。

有了以上程序，还要编译、存储，与 MC 联机传送程序，并存入 MC 的闪烁内存。此外，还要编写相应的 PLC 控制 MC 及必要时传送数据的梯形图程序，再下载给 PLC。这里有关梯形图程序略。

5.6.3　运动控制器

运动控制器是专用于运动控制的 PLC。OMRON 的 FQM（Flexible Quick Motion，柔性快速运动）就是其中一种，它有自身的协调模块，还有运动控制模块，可以用以创建 2 到 8 轴的柔性、高速、高精度的运动控制系统。FQM 还可扩展接入 CJ 系列的 I/O 或其它特殊 I/O 模块。也可实现一般 PLC 的功能。

1. FQM1 配置

（1）基本配置如图 5-110 所示。由一个电源单元、一个协调模块、一个或多个运动控制模块和一个终

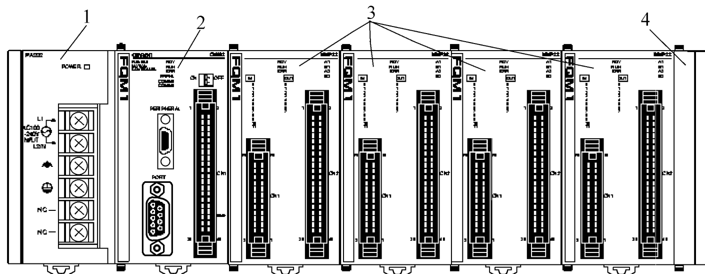


图 5-110　FQM1 系统配置

1—电源　2—协调模块　3—运动控制模块　4—终端模块

端模块组成。协调模块实际是 PLC 的 CPU，用梯形图程序实施对一般 I/O 控制及管理全系统运行。每个运动控制模块也有自己的梯形图程序，独立控制高速脉冲或模拟量输出。

1）FQM1-CM002 协调模块。内置有 16 点开关量输入、8 点开关量输出，32K 字的 DM 区。有 10K 步（相当于字）的程序容量的存储器，可用以存储梯形图程序，以协调运动控制模块的工作。同时，可实现 PLC 的其它功能。还有外设口和 RS-232 口，可用以与计算机或人机界面通信。此外，协调模块有一个循环与扩展循环刷新区。每个运动控制模块在此两区域中，分别分配 10 个字与 50 个字，在每个循环周期中被刷新。协调模块有一个同步数据链接位区，每个运动控制模块在此区域中分配 4 个字共享同步数据链接位区。

2）运动控制模块。有两种型号，FQM1-MMP22 与 MMA22。相当于小型 PLC。有自己的基本 I/O 点，有运动控制输出、输入点，有程序及数据存储区，可独立编程，直接进行运动控制。而各运动模块间的控制协调则通过协调模块及相互交换数据实现。运动模块的功能如表 5-35 所示。

表 5-35　运动控制模块功能

脉冲 I/O 运动控制模块	FQM1-MMP22	程序容量：10K 步 脉冲输入：2 脉冲输出：2 通用输入：12 通用输出：8
模拟量 I/O 运动控制模块	FQM1-MMA22	程序容量：10K 步 脉冲输入：2 模拟量输入：1 模拟量输出：2 通用输入：12 通用输出：8

3）电源单元。使用 SYSMAC CJ 系列电源单元。

4）FQM1-TER01 终端模块。一个 FQM1-TER01 终端模块随协调模块提供。在运动控制模块的最右侧安装终端模块。当使用了一个 I/O 控制模块后，在最后的 CJ 系列单元安装 CJ 1W-TER01。（在 I/O 控制模块中包括了一个 CJ1W-TER01 端板）。必须要安装一个终端模块，因为终端模块是机架的终结器。如果不安装终端模块就会发生一个致命 I/O 总线错误。

（2）扩展配置是为增加 CJ 系列机的模块配置。图 5-111 所示为扩展配置一例。

这里要使用 FQM1-IC101 I/O 控制单元。如果增加扩展机架还要用 CJ1W-II101 I/O 接口单元。这个 I/O 接口单元需要安装在扩展机架电源单元的右侧。

CJ 系列单元可以是基本 I/O 单元，也可是 CPU 总线单元，特殊 I/O 单元和通信单元，但有一些限制，具体见有关说明。

2. FQM1 功能

FQM1 可实现的功能较多。具体如表 5-36 所示。

3. FQM1 使用

（1）硬件配置。根据 FQM1 机功能表，按照系统使用要求，选定相关模块及驱动，做好硬件运动配置。

（2）按说明书要求安装硬件及接线。

（3）用 CX-Programmer 软件编程。建立的工程可包含一个 FQM1-CM 及 1 到 4 个 FQM1-MMP 或 FQM1-MMA。但 FQM1-CM 必须先加入。图 5-112 所示为所建立的 FQM 工程。FQM1-CM 在工程窗口的根节点上。其次为新 PLC2 为 FQM1-MMP，再就是新 PLC3 为 FQM1-MMA。这里共使用了 3 个 PLC。

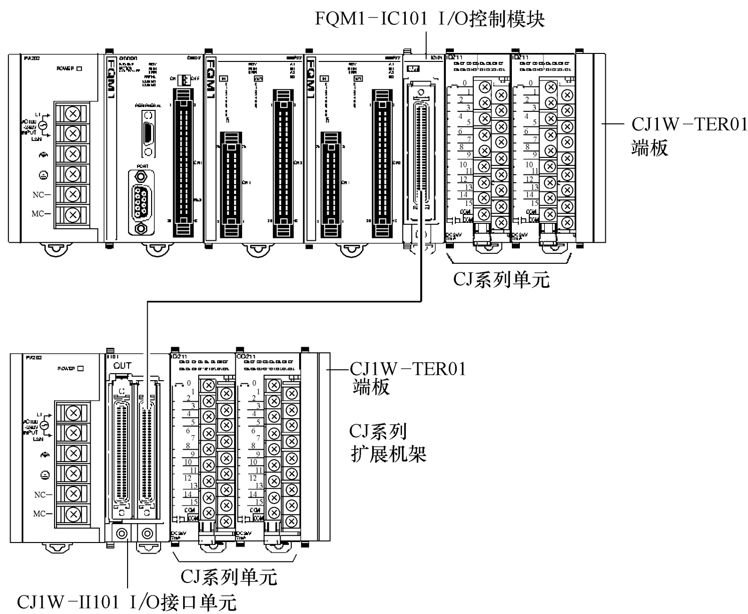


图 5-111　FQM1 机扩展配置一例

表 5-36　FQM1 功能表

控 制 分 类			应 用 示 例
运动控制	同步控制	滚刀	包装机器
		飞剪机	移动刀具
		电子凸轮	加工线和镜头加工
	直线控制	张力控制	卷绕和进给
		牵引控制	进纸
	扭矩控制	扭矩控制	注模
		扭矩限制	成形和挤压
	跟踪控制	CP 控制	加工和覆膜
		横贯控制	卷绕
测量控制	模拟系统	高速模拟量采样	纸张厚度检查和质量管理
		高速 PID 控制	定长距控制
	脉冲系统	高速计数器	测量(高速)和 F/V 转换
高速响应控制	I/O 控制	同步起动	传送机
		中断进给	贴标签机
		高速 PTP 控制	传送机
		高速计数器	传送机

由于它不支持 I/O 表、内存卡及时钟功能，所以，这些项目在工程中没有显示。每个程序只能由一个循环任务以及可多达 50 个的中断任务。

为了使 3 个 PLC 都能正常工作，必须按要求分别对其做好设定及编程。具体编程与本章 CP1H 机相同（含指令系统、内部器件）。编程后再下载给 PLC，并进行调试（本 PLC 系统不能进行仿真）。整个系统由

协调模块管理。工作模式，如处于编程、监控或运行，也是由协调模块决定。

（4）现场调试程序，直到满足要求。

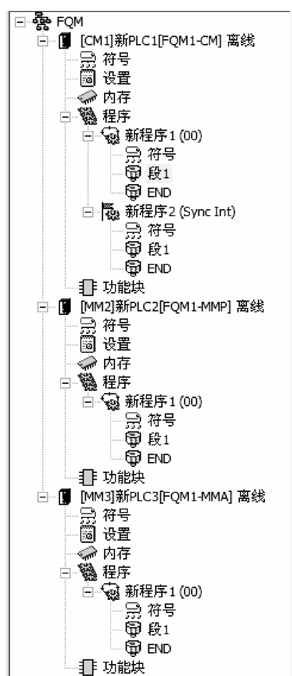


图 5-112 FQM 工程

结语

运动控制主要有两种，一是闭环，这多与高速计数功能配合，用开关量输出，常用于较简单的位移控制；另一为开环控制，较易实现多轴运动的插补运算，可以实现曲线运动。关键是 PLC 的运算速度及有相应的运算指令是否能满足要求。较好的方案是用专用位置控制或运动模块。

运动控制要用到脉冲量，但脉冲量还可用于过程控制，如使用 V/F（电压到频率转换）技术，可检测脉冲频率反映电压值，再用 PWM（脉宽调制）技术进行输出，则可以很简单、经济地实现闭环控制。

请想想

1. 脉冲量、脉冲量控制含义？它与运动控制的关系？
2. 脉冲信号产生、采集？
3. 脉冲输出类型、含义？
4. 脉冲量闭环控制的类型、特点及应用？高速计数比较控制的要点？
5. 单轴运动开环控制要点？
6. 双轴协调开环控制直线、圆弧插补算法要点及程序实现？
7. 双轴协调直接目标追踪控制算法要点及程序实现？
8. 列表数据控制程序要点？

请试试

1. 编写高速计数控制程序。
2. 编写简单脉冲量入、脉冲量（脉宽调制）出控制程序。
3. 设计正弦函数运动曲线控制程序。
4. 其它实际控制程序设计。

第 6 章　PLC 数据处理程序设计

一般讲，数据处理是指数据采集、存储、检索、变换、传输及数表处理。此外，有时还要考虑数据的录入与显示（输出）。

数据处理是很广泛的概念。前几章所讨论的种种控制，也可说是数据处理。如各种传感设备实时地收集被控对象的各种现场数据，加以适当处理和转换送入 PLC，PLC 对这些数据进行分析、判断、加工，进而产生输出，实现对系统的控制。这便是这里讲的数据处理。

但是，本章讨论的是 PLC 数据处理，其外延要小些。主要指用 PLC 采集数据、录入数据、存储数据、显示数据、传送数据及数表处理。本章将讨论与实现这个目的相关 PLC 编程。

6.1　数据终端是 PLC 的新角色

关键词：数据终端、专职数据终端、兼职数据终端

数据终端是指，具有以上讲的数据处理功能的基于计算机技术的设备。

早期，PLC 只用于控制，而且主要是逻辑量控制，无法用以存储数据。如 OMRON P 型 PLC，DM 区只有 64 个字。如启用了高速计数功能，还要占用 32 个字，可用的只剩下 32 个字。这么少的存储区存储不了多少数据，虽然它也是基于计算机技术的设备，但不能用作数据终端。

后来，随着 PLC 技术的发展，它的存储区也在扩大。如 OMRON H 型机，DM 区扩大到 2k，虽其中 1k 为特殊单元使用，但可用于存储数据的还有 1k。OMRON 新机型，这个存储区又增加到了几百 k 甚至更多。如考虑存储卡，则增加更多。正是如此，才可用其作数据终端。

这里所谓数据终端是指，它不用作控制，仅用作采集数据、录入数据、加工数据、存储数据、显示数据及传送数据。一般讲，用单片机或嵌入式计算机构成的系统多可担任这个角色。而有了足够存储空间的 PLC 也完全可以，而且还能更好地担任这个角色。

6.1.1　专职数据终端实例

以下举四个实例说明 PLC 是怎样担任数据终端这个角色的。

［**实例 1**］　沈阳华润啤酒厂用 C200H 机，进行灌装生产线数据采集。

沈阳华润啤酒厂某车间，有 5 条啤酒灌装生产线，两班倒。啤酒灌装的过程如图 6-1 所

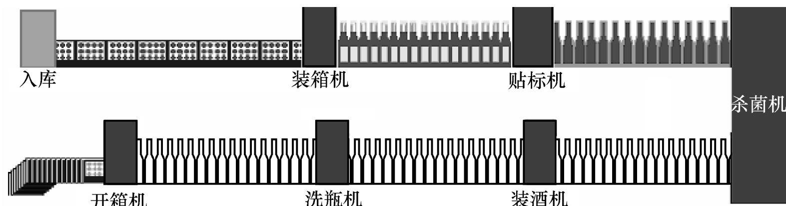


图 6-1　啤酒灌装生产线

示，具体是，送入装箱空瓶，开箱，清洗空瓶，装酒，加温消毒，贴标，装箱，入库。

1994 年，该厂配置 C200H CPU 21 型 PLC，专门对这 5 条装酒生产线有关数据进行采集、处理、显示、存储及传送，当作数据终端使用。

具体的数据有 3 组：

一是啤酒产量数据，各班次从管道流入的啤酒重量（用流量计检测）、空箱数、空瓶数、装酒瓶数、装箱数；

二是啤酒质量数据，各班次加温消毒是否超温或欠温的有关记录；

三是生产工作数据，各班次生产线何时开工、停工（含中间开、停工）有关记录。

以上记录的数据要在车间的数据显示屏上实时显示。同时，当计算机与其通信时，将按要求把这些数据传送给计算机，再由计算机作进一步处理与存储、报表打印及显示。构成了完整的啤酒灌装生产线的数据管理系统。

像这样用 PLC 作数据终端，成本虽高，但非常可靠。而且上手很快。这套系统除布线较麻烦，花的时间较长外，硬件安装、编程、调试仅半个月就完成了。

实践证明，有了这套系统，对该厂啤酒灌装管理、减少消耗及发挥各班次工人生产积极性都起到了很好的作用。并为实施该厂的 ERP（企业资源计划）积累了经验，建立了基础。

[实例 2] 东北电业局用 CQM1 型 PLC，对各变电所的用电量进行分时段采集与记录。

东北地区各变电所的用电量纪录，以前是不分时段的。而且都要就地查表并用手工记录。为了鼓励用电单位高峰少用电，低峰多用电，可否分时段记录，分段计价，而且能远程、自动地读取所记录的数据呢？答案是肯定的。

1994 年，东北电业局就针对这个问题作了研究。其解决方案是，上位机用个人计算机，下位机用单片机。下位机做数据终端，与每转可发送一个脉冲信号的电能表相连，就地采集这脉冲信号，逐日分时段累计这脉冲数，并予以存储。上位机在远程，不定时地（其时间间隔，要确保下位机旧数据不被新数据冲掉）与下位机，通过串行接口、MODEM、长途电话线路通信，读取所存储的数据，并进行处理、归档、显示及打印。投资几十万，研制一年多，取得初步效果。但是否可靠，不太有把握，故难以投用。

1995 年，正好 OMRON 在中国推出 CQM1 型 PLC。它的 DM 区有 6K 字，而且是模块式的小型机，其 CPU 还内装有 RS-232 通信口。经沈阳鹭岛公司推荐，并协助进行 PLC 编程，东北电业局科技处决定，在进行单片机系统测试的同时，用 CQM1 型 PLC（仅用 CPU 及内装的 16 个输入点，电源）作数据终端，上位机用 BORDLAND C ++ V3.1 编程，两个方案同时上。具体系统方案如图 6-2 所示。

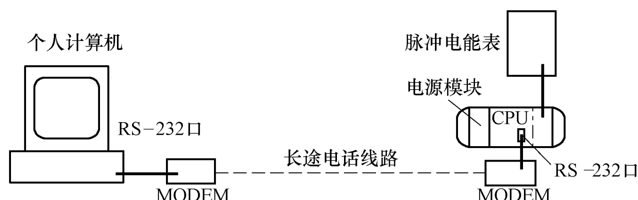


图 6-2 电量采集

实践证明，用 PLC 做数据终端的方案，单机成本虽然高，但开发周期短，开发费几乎是零。先后不到几个月的时间，就把辽宁地区的几十个变电所都作了改造。实现了在东北电业局调度大楼通过长途电话（通过 MODEM，并用计算机自动拨号）接通各变电所的 CQM1 机，读取经采集、存储在其中的各日各时段的用电量的数据。而这时，单片机作终端的方案，还在作有关测试，论证是否可以投用。

多年工作实践还证明，PLC 方案比单片机方案要可靠得多，做到了毫无差错，万无一失。

进入 21 世纪后，情况有了新变化。用户嫌 RS-232 串行接口通信速度太慢。拨号虽是自动的，但还是麻烦，费用也不少。要求用速度要快得多的，通信又非常方便且费用又较省的互联网读取数据。而 CQM1 型 PLC 没有以太网模块（OMRON 新推出的小型 PLC CJ1 有此模块）。所以，他们只好改用近几年才发展起来的带以太网卡的嵌入式计算机做数据终端，对系统作了更新。当然，如果 CQM1 有廉价的以太网模块，或 CPU 内装的以太网接口，或当时就有 CJ1 型 PLC，情况可能将有新的变化。

［**实例 3**］　辽阳钢管公司用 CPM2A 型 PLC，进行钢管保压性能数据检测。

辽阳钢管公司是上世纪 90 年代发展起来的民营生产有缝钢管公司，产品用于输送石油或天然气。早期，该公司生产的钢管只做 X 射线检查，未作钢管密封保压测试。2000 年，为了产品出口，对方要求进行这个测试。并要求用计算机作数据记录。

保压测试用耐压测试机。先把水装入密封的钢管中。然后加压，要加到 10 几个大气压，有的要加到 30 个大气压。接着停止加压（保压）。这时测试其压力变化。一般要求这个压力能保持 15 ~25s。

钢管尺寸很大，长 10 几 m，直径几百、几千 mm。测试机更大。测试机只能安装在生产现场，信号易受干扰。所以，只好用 CPM2A 型 PLC 在测试机现场，实时采集压力数据，然后安装在控制室中的个人计算机，通过串行接口读取存于 CPM2A 中的数据，进一步再作数据处理、显示、归档及打印。图 6-3 所示为该测试系统的简况。

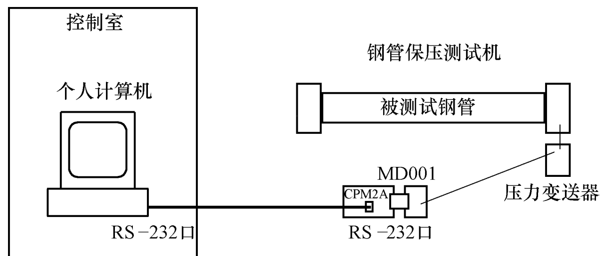


图 6-3　钢管保压性能数据检测

从图知，CPM2A 型 PLC 加一个模拟量扩展模块（MD001），用以接收从压力变送器送来的标准信号（4 ~20mA）。而压力变送器则实时检测被测试钢管的水压。

实际使用说明，此解决方案可行且工作可靠。2002 年，为适应产量增加的需要，按原方案又增建了第二套系统。

［**实例 4**］　沈阳三泰轮胎公司用 C200HE 型 PLC，对 27 台轮胎硫化机的合、开模时间、工作温度进行实时记录。

2002 年，沈阳三泰轮胎公司为了加强对生产管理，用一台 C200HE 型 PLC 与 27 台轮胎硫化机控制用的 C200H 型 PLC 相连，建立 PLC 链接网（PLC LINK NET），通过数据链接，采集各硫化机的合、开模时间、工作温度等数据。图 6-4 所示为该系统的简况。

从图知，各 PLC 都配备一个 PLC 链接模块（C200H LINK 401）。然后，用双绞线把所有这链接模块连接起来，组成 PLC LINK 网。

该网共有 28 个节点。用 LR 区实现链接。每个 PLC 设不同站号，0 到 27。对应占用不同的两个 LR 区的可写字。如站号设为 0，则可写字为 LR0 及 LR1 两个字。再如站号设为 10，则可写字为 LR20 及 LR21 两个字。余类推。

各个硫化机都采集本机的工作温度数据。并把此温度数据及本机合模、开模信号写在自

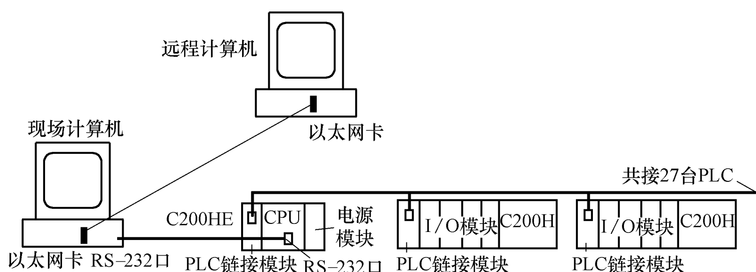


图 6-4　轮胎硫化机实时记录

己可写的 LR 字中。C200HE 型 PLC 定时读取、处理 LR 区的全部数据（通过链接通信，各站数据都集中于此）。并分历史与实时两类存储，以备现场计算机读取。

现场计算机与 C200HE 型 PLC 通过 RS-232 口通信，读取存储于 C200HE 机中的实时数据，并予以显示。同时，每天三次自动读存于 C200HE 机中历史数据（也可多次读），并进行处理、归档。

远程计算机通过以太网卡及其连线与现场计算机通信，读取有关数据，并予以显示、打印。

使用这个系统，可清楚地知道：每台硫化机何时合模，何时开模，进而统计各硫化机的实际产量（每 45 分合、开模一次，出产两个轮胎）；弄清合模时间是否过长或过短，而影响产品质量；弄清开模时间是否过长，而影响工作效率；弄清各硫化机工作温度有否超温或欠温，如超温或欠温，发生在什么时刻，有多长。所有这些数据，都以计算机文件的形式存储。并随时可在现场计算机或远程计算机上查看，打印。

显然，有了这个系统，该公司在生产信息化上前进了一大步。为加强生产管理与产品质量控制打下坚实的基础。

6.1.2　兼职数据终端实例

在 PLC 实施控制的同时，如果需要，也可用 PLC 实施数据采集、处理、存储、显示及传送。可更好地发挥 PLC 的优势。这也就是 PLC 兼做数据终端。以下也举四个实例说明 PLC 在这方面的应用。

［**实例 1**］　啤酒发酵温度记录。

啤酒糖化后的酒浆要送入发酵罐，进行发酵。一般要经近 30 天的发酵，才能成为啤酒。发酵过程酒浆会发热。但温度太高，酒浆可能变质。为此，要用干冰冷却。而温度太低，发酵过程也无法实现。所以这个发酵过程要进行温度控制。

要求的温度要按时间逐步下降。不同品牌的啤酒，有不同的温度下降曲线。如图 6-5 所示。

用 PLC 控制发酵温度，其目的就是使发酵能按温度下降曲线进行。温度高时，多加干冰，低时少加，或不加干冰。

为了验证发酵过程的温度确是按要求的温度下降曲线进行，车间的仪表员，每隔两个小时要记录一次酒浆的温度。这也叫抄表，该车间有 40 个啤酒罐都要抄表。显然，这是一个细致而又繁琐的工作。而且所抄表的表格要存档，档案要保留几个月。当啤酒质量出现问题

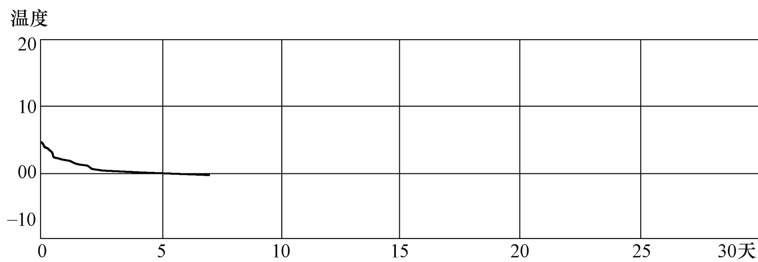


图 6-5　啤酒温度下降曲线

时，要查此档案做质量分析。据讲，该车间曾专设一个房间保存此档案。在那么多的档案中，要查某年某月某罐的档案也是一件不易的事。

为了抄表准确，减少手工劳动，减少文字材料以及存档、查档案的方便，该车间于 1994 年对 PLC 硬件进行改进。将原来用的 C200H 型 PLC 的 CPU 改为 21，内存也改为支持有内部时钟的型号。同时，程序也重编，以作到不仅可控制 40 个罐的温度，而且可采集这 40 个罐温度，并顺序存储这数据。只要上位机于三五天内把此数读走，所存的数据就不会丢失。

上位机程序是用 VB 编写的，可读取、处理、显示所有数据，并以文件形式存档，也可存入 ACCESS 编的数据库，并可按需要打印。

这套系统经多年运行说明，当初的设想是可行的。确是达到了抄表准确，无需手工劳动，减少了文字材料，以及存档、查档案方便的目的。

［**实例 2**］　辽宁义州市某水果园各灌溉点的用水量统计。

1997 年，辽宁义州市某果园引进以色列节水灌溉技术，用 CPM1A 型 PLC，控制 16 个果园小区的灌溉。同时，要统计各小区累计供水时间及累计供水量，以便收费。所以，在 PLC 编程时，不仅考虑有关控制问题，还要考虑有关数据采集、处理及存储问题。

图 6-6 所示为上位机显示的画面。而上位机显示的数据，则是从读取存于 PLC 中的有关数据得来的。

1998 年，沈阳苏家屯李英乡果园也用相关技术。它仅有 4 个果园。但控制、记录与义州市是一样的。它用的 PLC 也是 CPM1A。

显然，为了向上位计算机提供有关数据，CPM1A 型 PLC 就要担负起数据终端的角色。

小　区	累 计 时 间	累 计 水 量
小区 01		
小区 02		
小区 03		
小区 04		
小区 05		
小区 06		
小区 07		
小区 08		
小区 09		
小区 10		
小区 11		
小区 12		
小区 13		
小区 14		
小区 15		
小区 16		
合　计		

图 6-6　显示数据

［**实例 3**］　阜新液压件厂油泵冲击实验时压力波形记录。

阜新液压件厂是我国重要的叶片泵生产及开发厂。它的实验室要作各种新油泵的种种测试。其中一项为冲击实验。其测试工艺是，油泵正常运转后，突然要关闭出口。这时，泵的油压急剧上升，并且伴有抖动。冲击实验就是要造成这个抖动，并测试这时压力的变化（波形）。

因为测试整个过程仅几秒钟，每隔几毫秒就要测一次压力数据，还要避免现场强电的干

扰，所以，用 CPM1A 型 PLC 的 MD001 模块采集压力信号。并每隔几毫秒采集这压力数据，并在 DM 区相继的 400 个地址中存储这个数据。

PLC 中有了这个数据，当上位计算机与其通信时，即可读取这个数据，并加以显示、分析及存档。这里，PLC 的数据终端作用是不可或缺的。

该厂还在同样的系统中，用外中断进行被测试油泵转速的脉冲量采集。用采集到的脉冲频率，换算出油泵的转速。

［**实例 4**］ 辽源汽车油泵厂油泵测试数据记录。

汽车油泵产量很大。而且出厂前必须对其有关性能进行测试。要检查在额定转速下，它的流量、压力及转矩。

2001 年，辽源汽车油泵厂从德国引进不用油液的油泵测试机，但控制系统自行设计。其解决方案是，一台上位计算机，两台测试机。这两台测试机都用 CPM2A 型 PLC 控制，并记录以上三个数据，占三个 DM 字。

在开始测试记录此数据前，还要用手工输入件号。件号占两个 DM 字，也要存储，以区别是不同泵的测试数据。

手工输入件号输入用小键盘。通过 PLC 的输入点送入相应 DM 的两个字。可知，这时的 CPM2A 型 PLC 还是一个录入终端。

测试后，每个泵占五个 DM 字记录。

CPM2A DM 区有 2000 个字，故最多可存储 400 个泵的测试数据。但实际上还要少些。因为，还要有一些 DM 字用作控制。

由于测试机可存储这么多数据，故两台测试机可先单独作测试与记录。到记录足够数据后，再与上位计算机联机，把数据传给上位机，并清除已存数据。传给上位机后，上位机再作进一步处理、存档及打印。

显然，如果不是 CPM2A 型 PLC 用这 2000 字的 DM 区兼作数据终端，这个方案是不能实现的。

从以上的 8 个实例的介绍可知，PLC 用作数据终端，或兼作数据终端，已是 PLC 应用的一个重要方面。随着在信息化基础上的自动化的发展，PLC 这个应用将越来越多。

用 PLC 作数据终端的好处是，做数据终端的同时，可用作控制。抗干扰能力强，实时性好，可靠性高，体积小，可在工业现场直接采集数据。同时，用 PLC 作数据终端，开发费用小，开发周期短，很快就可投入使用。此外，PLC 联网能力强，可通过各种网络途径使其成为某计算机的数据终端。

6.2 数据终端条件及其使用

关键词：数据存储区、扩展数据存储区、内存卡、指针、偏移传送、寄存器访问、使用〖栈访问

PLC 能起到数据终端的角色，要有相应的条件。最主要的是 PLC 要有足够的数据存储区。OMRON PLC 称之为 DM，而有的 PLC 厂商称变量区（V）。从以上介绍的 8 个实例知，没有足够大的 DM 区，用 PLC 进行大量数据的存储是不可能的。

扩展数据存储区，OMORN 称之为 EM 区，是 DM 区的备份、补充及扩展。其容量一般

比 DM 区大。可与 DM 区配合，使存储的数据成倍的扩大。

内存卡是可移动的存储介质。存储量更大。由于是可移动，有多少卡，就可以存储多少数据。所以，也可说它是“海量”存储。

数据采集总是与时间相联系的。总是要记录采集当时的具体日期、时刻。作其它处理时不少也要用到时间。所以，PLC 要用作数据终端，一般要有实时时钟。当今 PLC 多都有实时时钟。如无此时钟，6.2.4 节提供有时钟程序，可供参考。

此外，PLC 还应有相应的联网能力。这也是作为数据终端必不可缺少的条件。有关问题将在本书第 7 章讨论。

6.2.1　DM 区及对其访问

DM，数据存储区的简称，是 PLC 基本的存储介质。它容量大，又为掉电保持的，故可用以大量而长久的保留所存储的数据。

OMRON DM 区按字使用，以字编址。在 CJ1 型 PLC 之前，不能对字中的位进行处理。有的 PLC，如西门子 S7-200 型 PLC，数据存储区不称 DM，而称 V 区，以字节编址（VB）、使用，但也可按字（VW）、按双字（VD）使用（仍按字节编址）。也有的称为 D 区。

除了 DM、V，PLC 的其它内部器件也可用以存储数据。有的也可或也是掉电保持的。只是这些器件的容量要小些。

有了 DM 区或其它存储器件，程序如要直接访问它，指明地址就可以了。只是直接访问不够灵活，效率也不高。要能灵活地访问，不用很多指令就能频繁地访问，就得用间接访问。间接访问有 4 个方法：

1. 使用指针访问

OMRON DM 区可用指针实现这个间接寻址。指针是 C 语言常用的编程技法，正确使用指针可提高的程序效率。但使用前一定要控制好指针。失控的指针，也就是间接地址不存在的或不希望出现的指针称野指针。使用这样的指针是危险的。

图 6-7 所示为使用指针传送数据的程序。

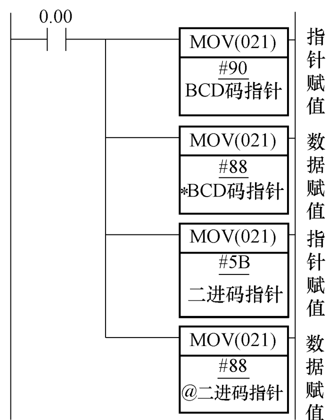


图 6-7　用指针传送数据

从图知，使用指针访问，先总是指针赋值，再才是数据赋值。数据赋值时，把指定的数据传送给指针指向（指针的值）的地址。

早期 OMRON 公司 PLC DM 区不大，用 BCD 码指针（4 位最大为 9999）足以访问整个 DM 区。而今，有的 PLC DM 区可大到 32K，只得用二进制码指针。16 位二进制码指针最大可为 66735，足以访问所有的地址了。

图 6-7 所示的程序，在两种指针的赋值与使用上是有区别的。用 BCD 码指针的那两条指令，是把即时数 88 赋给 DM90。用二进制码指针的那两条指令，是把即时数 88 赋给 DM91（16 进制码 5B，即 91）。

西门子 PLC，如 S7-200 型，也有间接寻址功能。不同是它的指针可指向任何内部器件，而且是所指向器件在 PLC 内存中的实际（绝对）地址。所以，它的指针是双字长的，而且，它的指针以字节为单位，而不像 OMRON 那样以字为单位计算。

以下几个语句表语言表达的程序，就是使用有关指针的例子。

```
MOVD   &VB10   VD500        // 双字传送，把 VB10 的实际地址赋值给 VD500
ADDI   +1   VD500   VD500   // VD500  加 1
MOVB   VB0   *VD500         // 字节传送，把 VB0 的值传送给 VD500 指向的地址，即 VB11。
```

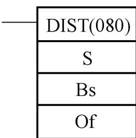
提示：西门子 PLC 程序，如要控制指针，则多是另用一计数器实现。

用指针访问 DM 区时，常要修改指针，并要对指针值进行控制，以免指针超出范围，这些将在本章以下各节还要作进一步讨论。

2. 使用指令作偏移传送

OMRON 有两个指令：DIST（080）及 COLL（081）。也可作与使用指针类似地访问 DM，甚至访问其它内存区。可用它作偏移传送。其功能与上述指针类似，而且，还不仅可处理 DM 区，还可处理其它数据区。

DIST（080）指令的梯形图格式为

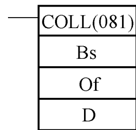


这里 S——原数据字。

Bs——目标字基址；

Of——目标字偏址。

COLL（081）指令的梯形图格式为：



这里 Bs——原数据字基址；

Of——原数据字偏址；

D——目标字地址。

图 6-8 所示为使用这两条指令把 DM0 到 DM99 的数据倒序传送到 DM100 到 DM199 的程序。

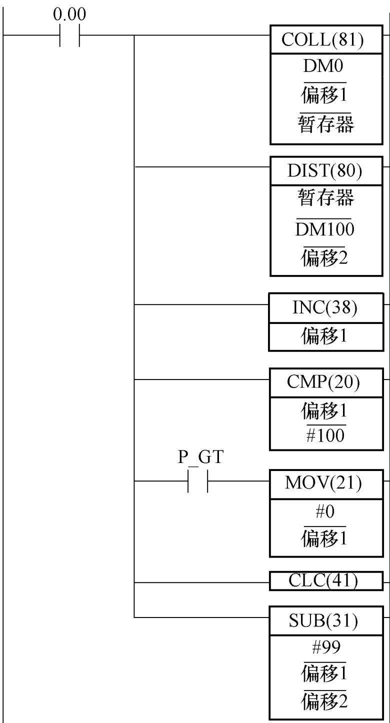


图 6-8　用偏移指令访问

由图 6-8 可知，它有两个偏移，偏移 1 与偏移 2，且两者一个增时，另一个则减，两者的和为 99。这两个偏移就相当于两个指针。偏移 1 用于从不同地址取数，偏移 2 用于向不同地址传数。每次取、传数后，也修改偏移。这样，一次次执行这程序，将实现上述目标。

此两指令，不仅可指向 DM 区，也可指向任何数据区。

3. 寄存器访问

高档的 PLC 多有变址寄存器及数据寄存器。如 OMRON 的 CS1 型 PLC 及 CJ1 型 PLC 即有此寄存器。变址器寻址可把间接寻址扩大到其它数据区，而且，还可处理位（BIT）数据。使数据的使用可更加灵活。

图 6-9 所示为使用变址寄存器完成与图 6-8 功能完全相同程序。

从图知，它的取数、传数，修改指针，仅用一条 MOV 指令。但操作数用的是变址寄存器（指针）指向的地址。IR0 为取数后修改其内容，IR1 为先修改内容后传输。图中用的其它指令都是为了处理指针的。有指针初始化，当开始执行本程序或传送完成时，对指针初赋值。指针控制，用一个计数器控制指针。

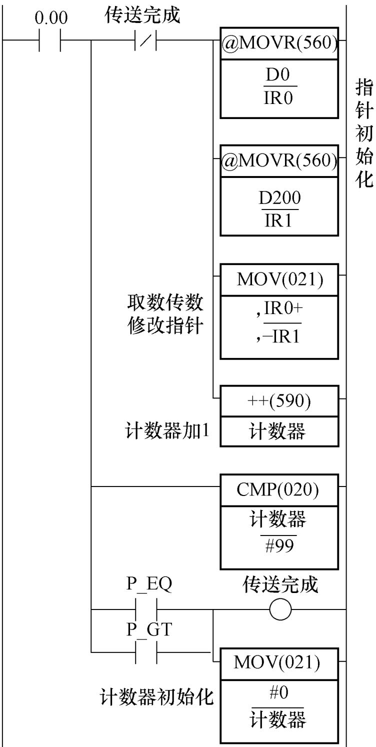
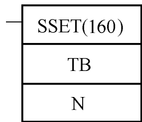


图 6-9　寄存器访问

4. 使用堆栈访问

此外，高档的 PLC 在系统的其它数据区中，还可建立堆栈（先进后出存储区）或队列（先进先出的存储区）。对此也设有相应的管理指令。

（1）设堆栈指令 SSET（160）。其梯形图格式为



这里 TB——栈的首地址。TB 及 TB＋1 的内容为栈的最高地址，TB＋2 及TB＋3 的内容为栈的指针；

N——栈的总长度，以字为单位。

如 TB 为 DM00000，N 为常数 10，执行本指令，则建立一个栈顶为 DM00000，栈底为 DM00009 的长度为 10 个字的堆栈。其初始化指针指向 TB＋4。图 6-10 所示为所建堆栈示意。

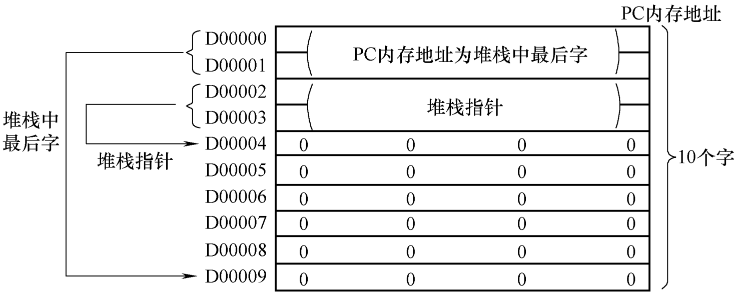
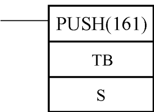


图 6-10　建堆栈

建栈之后，可接受堆栈的进栈、出栈指令操作。

（2）进栈指令 PUSH（161）。其梯形图格式为



这里 TB——栈的首地址；

S——源数或源数地址。

执行本指令过程是，将把源数写入堆栈指针指出的地址中，然后堆栈的指针加1。图6-11所示为进栈示意。

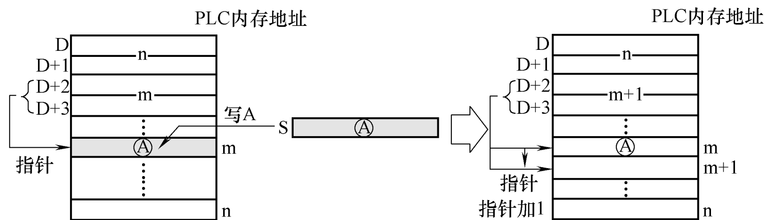
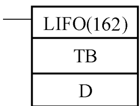


图6-11 进栈示意

（3）出栈指令。有两个，先进后出LIFO（162）及先进先出FIFO（163）。

1）LIFO梯形图格式为



这里 TB——堆栈的首地址；

D——目标数地址。

执行本指令过程是，先堆栈指针减1，然后从指针指向的地址中读出数，再写入D指出的地址中。图6-12所示为LIFO示意。

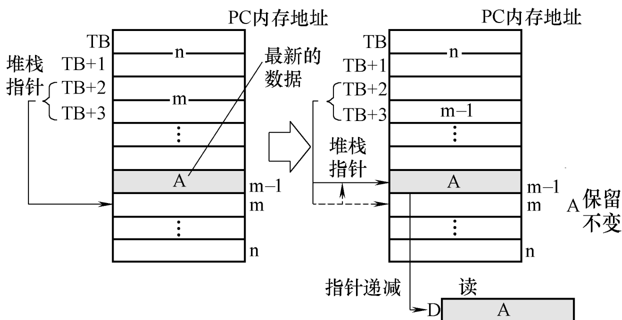
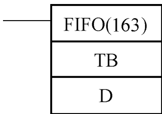


图6-12 LIFO示意

2）FIFO梯形图格式为



这里 TB——堆栈的首地址；

D——目标数地址。

执行FIFO指令过程是，从栈顶（地址TB+4）取数，字送目标地址D，然后栈指针（TB+3和TB+2）减1。栈中余下的数据向前串一个字，TB+4中原数据被新数取代。图6-13所示为FIFO示意。

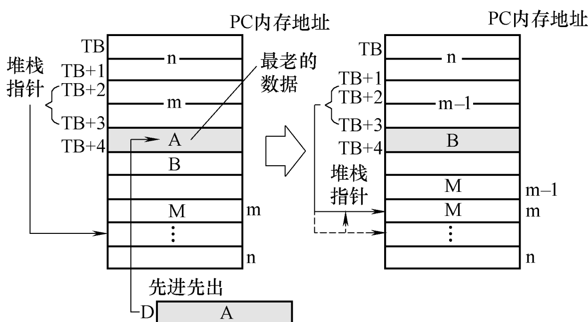


图6-13 FIFO示意

图6-14所示为以上堆栈指令应用程序。

从图知，它的作用也与图6-8的

作用相似。也是把 DM 一个区的数据倒序传送到 DM 另一个区。只是它还要留 4 个字用作堆栈的管理使用。

6.2.2　EM 区及对其访问

扩充数据存储区，即 EM，高档的及新近的 PLC 才有这个区。而一般的 PLC 仅有 DM 区。EM，指扩展数据存储器，少的 6k，多的可扩到 24k，32k……。

由于 PLC 地址总线宽度的限制，对多于 6k 的 EM，只好对其作分段管理。每段约 6k，新的 PLC 较大。

CS1H/CJ1H 型 PLC，其 EM 区与 DM 区访问是相同的。如传送指令（MOV）也可对 EM 区进行操作。如图 6-15 所示，当 0.01 位 ON 时，即时数 88 将传送给以指针 E0_0 的内容（BCD 码）为 EM0 段地址的字中；即时数 888 将传送给以指针 E0_1 的内容（二进制码）为 EM0 段地址的字中。

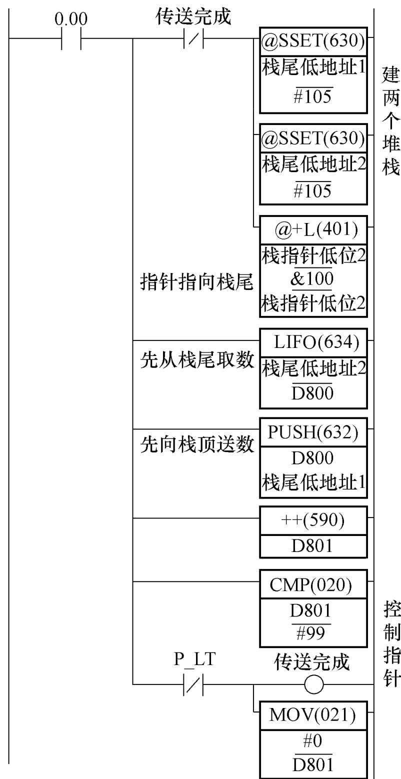


图 6-14　栈指令应用程序

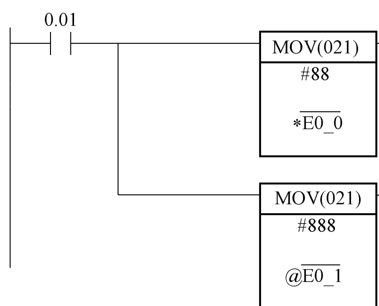


图 6-15　EM 区访问

但在此之前，PLC 的多数数据处理及逻辑处理指令仅适用于 DM，无法对 EM 区作操作。为此，设置了 EM 的 PLC 增加一些能对 EM 区进行操作的指令，以及两个区转换的指令。

6.2.3　内存卡及对其访问

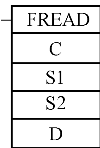
内存卡主要用于程序备份。但现在新型 PLC，也可用以与数据区的交换数据。数据区的数据转存到内存卡后，又可存新的数据。这无形中等于扩大 PLC 数据区的存储容量。

而且，通过 PLC 内存卡适配器，内存卡还可单独与计算机联机。计算机可以文件的形式读写内存卡。这为 PLC 脱机与计算机交换数据也提供了方便。

为使用内存卡，PLC 还提供了相应的指令。如读数据文件、写数据文件等。

1. 读数据文件：FREAD（700）

用以从文件内存的指定数据文件中，读取指定的数据，并存放到 CPU 单元中指定的数据区中。其梯形图符号为



这里　C——指明源文件是在存储卡，还是在 EM 文件内存中；

S1——指明是读取实际数据还是读取数据的字数；

S2——指明回车是否存在；

D——指明数据类型。

执行 FREAD 指令，则从 S2 中指定的文件（文件扩展名为 .IOM）中读取 S1 和 S1 +1 中指定的字数，S1 +2 和 S1 +3 中指定的起始地址。然后将数据写入以 D 指定的开头的内存区中。

例：程序如图 6-16 所示。这里，C 为 0000，含义为从内存卡读数据。S1、S2 中相应设定，如图 6-17 所示。当 0.00 ON，FREAD 从文件\ABC\XYZ.IOM 文件中，读取以文件起始 +5 个字开始的 10 个字的数据，并将这 10 个字输送到 D400 ~ D409 中。图 6-18 所示的为程序执行读数据的情况。

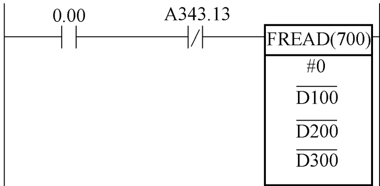


图 6-16　读文件程序

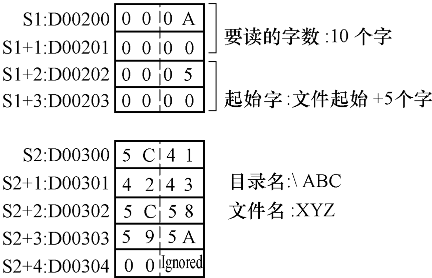


图 6-17　S1、S2 区设定

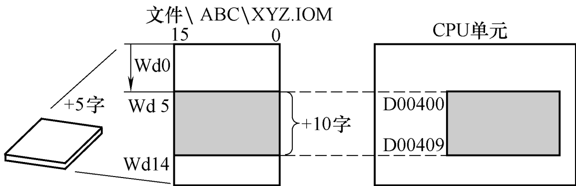
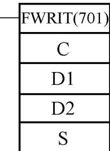


图 6-18　读数据情况

2. 写数据文件：FWRIT（701）

用以把数据区中指定的数据，添加或重新写入文件内存的指定文件中。如果指定文件不存在，将产生该文件名的新文件。数据可以是二进制、文本或 CSV 格式。其梯形图符号为



这里，　C——控制字；

D1、D1 +1——写入项的数；

D2——文件名；

S——源首字。

有 3 种情况：

（1）在已有文件中重写数据（C 的第三个数为 1）。如图 6-19 所示，在 S 中，指定数据区的起始字。在 D1 和 D1 +1 中，指定重写字数或字段数，在 D2 中，指定文件名（文件扩展名 .IOM，.TXT，.CSV），在 D1 +2 和 D1 +3 中，指定起始地址。

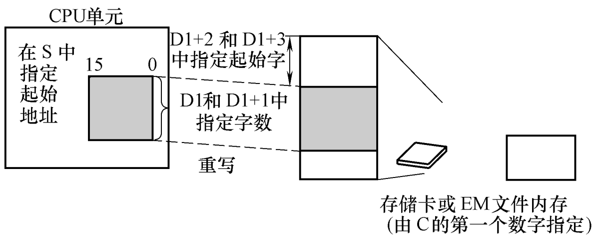


图 6-19　在已有文件中重写数据

（2）在已有文件中添加数据字数（C 的第三个数为 0）。如图 6-20 所示，在 S 中，指定数据区的起始地址。在 D1 和 D1 +1 中，指定添加字数或字段数。在 D2 中，指定文件名（文件扩展名为 .IOM，.TXT，.CSV）。在 D1 +2 和 D1 +3 中，指定起始地址。

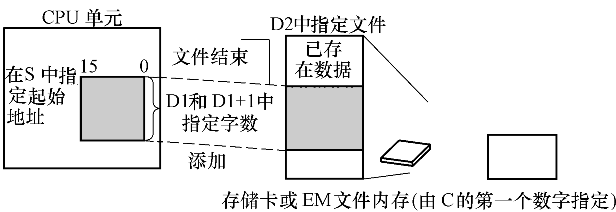


图 6-20　在已有文件中添加数据

（3）用源数据创建一个新文件。如图 6-21 所示，如果 D2 中指定的文件不存在，FWRIT

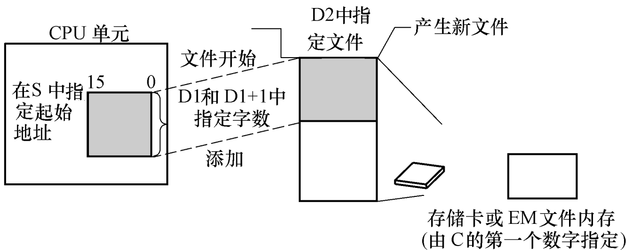


图 6-21　用源数据创建新文件

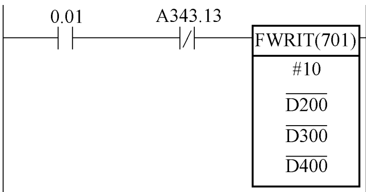


图 6-22　写数据文件梯形图程序

C	#0 0 1 0	文件内存:存储卡 功能:重写数据
D1:D00200	0 0 0 A	要写的字数:10 个字
D1+1:D00201	0 0 0 0	
D1+2:D00202	0 0 0 5	起始字:文件起始 +5个字
D1+3:D00203	0 0 0 0	
D2:D00300	5 C 4 1	目录名:\ABC
D2+1:D00301	4 2 4 3	文件名 :XYZ
D2+2:D00302	5 C 5 8	
D2+3:D00303	5 9 5 A	
D2+4:D00304	Ignored	

图 6-23　D1、D2 设置

将产生一个具有该文件名和扩展名（.IOM，.TXT，.CSV）的新文件，并将指定的源数据以指定的文件类型，从文件起始写入。与是否指定添加重写数据无关。

例：写数据文件梯形图程序如图6-22 所示。C 的设定值为 0010，含义为向内存卡重写数据。D1、D2 中相应设定，如图 6-23 所示。当 0.01 ON，从 D00400 ~ D00409 中读 10 个字的数据，并将该数据重写到文件\ABC\XYZ.IOM 中以文件起始 +5个字开始的 10 个字中。图6-24所示为写数据文件程序执行情况。

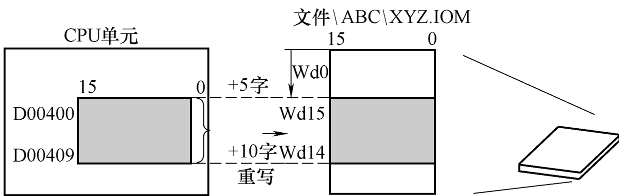


图 6-24 写数据文件程序执行情况

6.2.4 时钟程序

图 6-25 所示的为时钟程序。执行本程序后，CNT 105、CNT 104、CNT 103、CNT 102、CNT 101 及 CNT 100，将分别存储相应的当前的年、月、日、时、分、秒值。

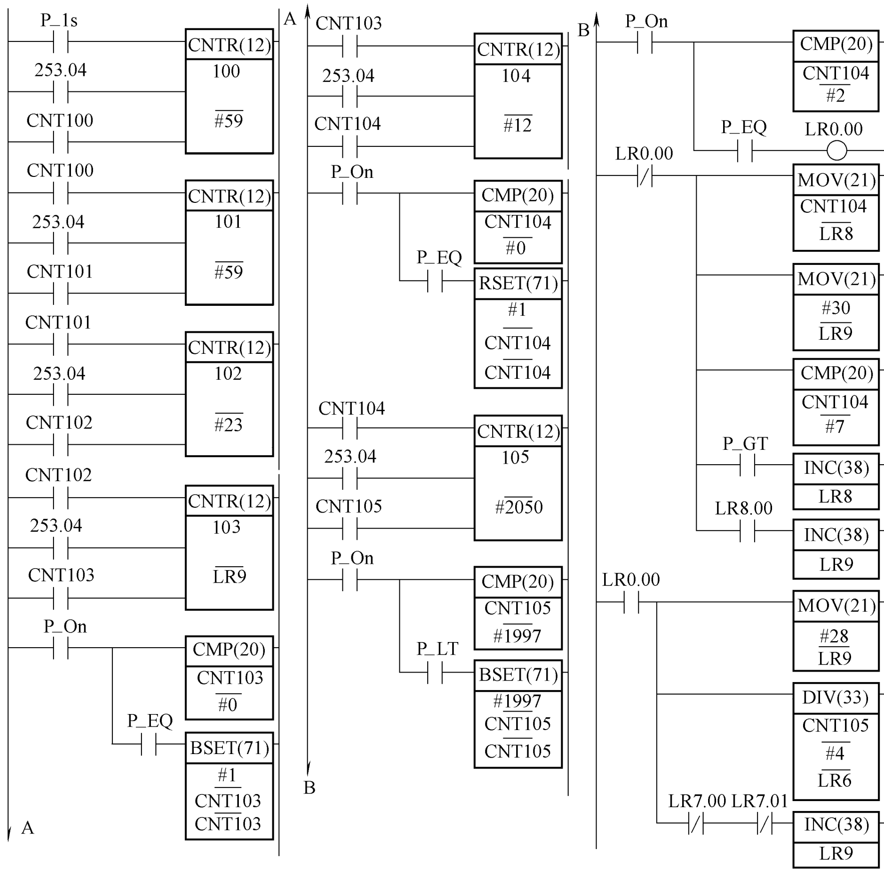


图 6-25 时钟程序

从图知，它有 CNT 100 ~ CNT 105 6 个可逆计数器，其减计数器接常 OFF 接点（即25314），而增计数端，CNT 100 的接秒脉冲，其余依次接其前级的计数器常开接点。以秒计

数器为例，当其计数值达到 59 后，再增一秒，则 CNT 100 复位，并产生进位脉冲，使分计数器 CNT 101 增 1。

分计数器达到 59 后，再增一分，则分计数器 CNT 101 复位，并产生进位脉冲，使时计数器 CNT 102 增加 1。

时计数器达到 23 后，再增一小时，则日计数器 CNT 103 复位，并产生进位脉冲，使日计数器增 1。

日计数器到底什么时候进位，要依大小月及是否为 2 月以及是否闰年有关。这里用 L9 的内容确定。大月时 L9 为 31，小月时为 30。正常年份的 2 月为 28，而闰年的 2 月为 29。闰年与否，依年能否被 4 整除而定（考虑到 1995 ~ 2050 年的情况）。图中用了一系列的比较指令，决定了 L9 的内容。

日计满后，向月进位，月满后向年进位。为保证日、月计数器的起始值为 1，各用了一个比较，只要它为零，则把它显成 1。因为月、日不像时、分，没有零值。

提示：执行 MOV 指令，也会改变标志位 P-EQ。当被传数为0时，P-EQ 置0，否则置1。

OMRON 的 CQ1M 型 PLC 内置时钟的值存于 AR18 ~ AR20 字中。依次是：AR18：低字节为秒，高字节为分；AR19：低字节为时，高字节为日；AR20：低字节的月，高字节为年。这里年仅两位数，97 为 1997。但若为 01，则为 2001 年。

为了便于使用，把上述计数后的年月、日时、分秒及当前的时分，分别用 4 个字显示。其程序如图 6-26 所示。

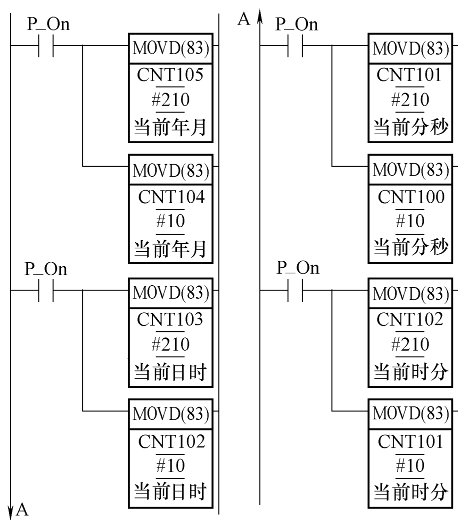


图 6-26　4 字显示时钟程序

这种时钟的缺点是，PLC 不工作，时钟也就不走了，只能用于 PLC 长期不停止工作的场合。如果 PLC 停止工作后又重新起动，这时要进行对时；可用编程器、操作器或上位计算机改变 CNT 100 ~ CNT 105 的现值实现。

这种时钟不可能太准，因为定时脉冲精度会受扫描周期变化的影响。一般都会变慢。有时，一天可慢 1 ~ 2 分钟。为了解决这个问题，最好用定时中断（如定时 1 秒一次中断）来执行这组程序，或至少执行其中的秒计数器，则可作到分秒基本不差。也可不用秒计数器，用分脉冲直接驱动计数器 CNT 101，这样计时误差小些，但时间不能以“秒”计。

6.2.5　时钟设定

目前大多数都内置有时钟。不必运行时钟程序，做好设定就可以使用内置时钟。设定的方法有：

1. 编程软件设定

运行 CX-Programmer 软件，联机。在其主窗口的“PLC”下拉菜单上，找到“编辑”项。再点击其下的“时钟”项如图 6-27 所示。将弹出“PLC 时钟 ”窗口，这时，如果点击

“选项”中的“同步时钟”按钮，则把计算机的时钟现值下载给 PLC。

如果另作设定，可点击窗口上的“选项”下拉菜单。再点击“设定 PLC 时钟”项，将弹出“设定 PLC 时钟窗口”，如图 6-28 所示。可在其上分别对时间及日期进行设定。

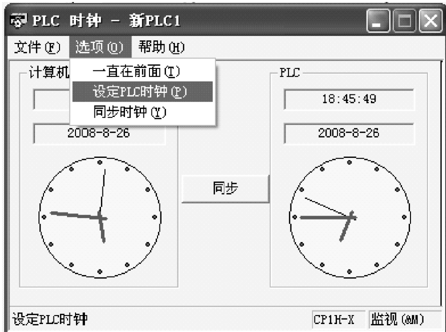


图 6-27 PLC 时钟窗口

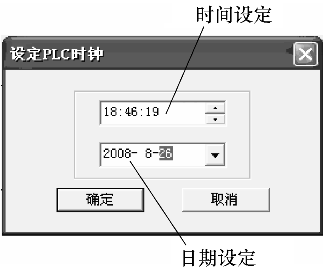


图 6-28 设定 PLC 时钟窗口

CP1H 型 PLC 时钟在辅助区 A351 ~ A354 中显示。如表 6-1 所示。

表 6-1 CP1H 型 PLC 时钟辅助区

地 址	内 容	地 址	内 容
A351.00 A351.07	秒(00 ~ 59, BCD)	A353.00 A353.07	月(01 ~ 12, BCD)
A351.08 A351.15	分(00 ~ 59, BCD)	A353.08 A353.15	年(00 ~ 99, BCD)
A352.00 A352.07	时(00 ~ 23, BCD)	A354.00 A354.07	星期(00 ~ 06 = 星期日到星期六)
A352.08 A352.15	日(01 ~ 31, BCD)	A354.08 A354.15	常为 00

2. 手持编程器设定

C 系列机型，以 CQM1 型 PLC 为例，其时钟显示辅助继电器为 AR18 ~ AR21。与 CP1H 型 PLC 的 A351 ~ A354 基本对应。可用手持编程器直接写这个区进行设定。办法是：

（1）先置 AR2114 位 ON，使时钟停止。

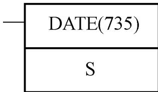
（2）按设定时间要求，修改 AR18 ~ AR20 的内容。

（3）当达到步骤 2 设定的时间时，置 AR2115 位 ON，时钟从这个设定的时间开始运行，并自动置时钟停止位 OFF。当完成时间设定后，AR2115 也将自动置 OFF。

只设定秒时，用 AR2113 位。可简单设定“秒”为“00”，当 AR2113 为 ON 时：如果秒是从 00 至 29，清除秒为“00”，分钟不变。如果秒是从 30 至 59，清除秒为“00”，分钟加 1。

3. 程序设定

CP1H 型 PLC 还可用程序设定。所用指令为 DATE。其图形图格式为



这里的 S、S + 1、S + 2、S + 3 与 A351 到 A354 对应，存放时间设定值。

图 6-29 所示为设定时钟程序。当 0.01 ON，将把 D1500、D1501、D1502、D1503 设定的时间下载给 PLC。如设定值如表 6-2 所示，则设定后的时间为 2008 年 8 月 8 日 8 时 8 分 8 秒，星期五。

表 6-2　D1500 设定

地　　址	值(BCD)	地　　址	值(BCD)
D1500	0808	D1502	0808
D1501	0808	D1503	0005

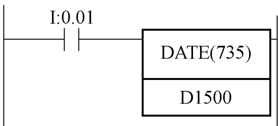


图 6-29　时钟设定程序

6.3　数据采集程序设计

关键词：开关量采集、模拟量采集、脉冲量采集、脉冲选通采集

数据采集可分为开关量采集、模拟量采集及脉冲量采集。而采集时一般总要与时间相联系，总是要指明采集当时的具体时刻和日期。以下分别对这些量的采集作说明。

6.3.1　开关量采集

开关量本身仅两个取值，较简单。如 ON 代表开工、OFF 代表停工。采集它的目的主要是，弄清什么时候开工，什么时候停工。图 6-30 所示的梯形图即为这个开关量采集程序。

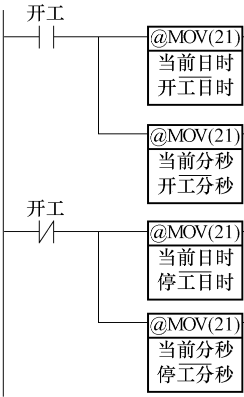


图 6-30　开关量采集

从图知，当“开工”信号 ON，则把“当前时日”传“开工日时”，“当前分秒”传“开工分秒”数据区中。而当“开工”信号 OFF，则把“当前时日”传“停工日时”，“当前分秒”传“停工分秒”数据区中。注意，这里用的都是“微分”传送，只是在 ON 或 OFF 的那个扫描周期才进行这个传送。

显然，上位计算机读取采集的这两组数据，稍作比较，就可清楚当前是开工还是停工。如是开工，还可知道是什么时候开工。以及上次是什么时候停工。

6.3.2　模拟量采集

PLC 的模拟量是从模拟量输入单元读取的。而且，这个读取时间的延迟是很短的。一般为 PLC 扫描周期级的。个别的如 C200H-TS001 之类温度检测单元要作一些平均数计算，为秒级。所以，模拟量输入通道有了新数时，也就完成了模拟量采集。

只是，模拟量输入单元所读的数据多为二进制数，而且所用的单位及初值也常不合显示或存储的要求。因而常要对这些数据作必要的转换。显然，加上这个转换，数据采集也就完整了。图 6-31 所示梯形图即为这样一个转换程序。



图 6-31　转换程序

该图程序用于 CPM1A_MA002 读入数据的转

换。因该模拟量输入单元读入的是 8 位二进制数，故一个模拟量输入通道读入的是两路数据，须把它分开。并还要转换为 BCD 码，该图程序所作的就是这个工作。它把模拟量输入通道 1 读入的数据分成“第一路二进制码输入值”及“第二路二进制码输入值”，然后转换成“第一路 BCD 码输入值”及“第二路 BCD 码输入值”。

有时还须把采集的数据与采集时间相连系，并用能看出被采集量的变化，即所谓变化趋势监视。这个工作一般由上位机去做。但 PLC 本身也可完成。

为此，可在 PLC 的某存储区设定一组（如 10 个字）工作区。用这个工作区动态记录被采集数据与采集时间有关的信息。

对此，有两种方法：一是定时采集；二是变化采集。

1. 定时采集

可按一定的时间间隔采集数据，并按固定的地址记录。定时采集因采集时间是固定的，可不必记下采集时间。图 6-32 所示即为这种梯形图程序。

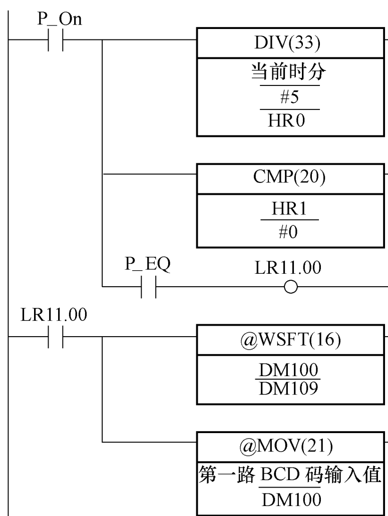


图 6-32　定时采集

从图知，这里先是把“当前时分”（存当前几时几分字）被常数 5（也可为别的常数）整除，其商数存于 HR0，余数存于 HR1 字中。然后再对 HR1 与常数 5 作比较。

如这时的时间为 5 分，或 10 分…，则比较相等（P-EQ ON），进而先把 DM100 ~ DM109 中的数按字移位，DM108 的数移给 DM109，DM107 的数已给 DM108，等等。然后把最新的“第一路 BCD 码输入值”存入 DM100。

可知，这里 DM100 ~ DM109 中存的数分别为记录当时及前 5 分、前 10 分…的被采集的数据。并每 5 分钟作一次更新。

2. 变化采集

即跟踪被采集量，视其变化情况，若变化值超过某个范围，则采集，并同时记下这时的时间。再有新的变化再采集。图 6-33 所示即为这种梯形图程序。

从图知，这里总是进行“第二路 BCD 码输入值”与“输入暂存值”相减，得其差的绝对值。然后把这个“差”与常数 5 比较。如比较大过常数 5（也可为别的常数），则 P_GT ON，进而 LR10.00 ON。接着，先把 DM200 到 DM209 中的数按字移位，DM208 的数移给 DM209，DM207 的数移给 DM208，等等。然后把最新的“第二路 BCD 码输入值”存入 DM200。再接着，又把 DM200 ~ DM209 中的数按字移位，DM208 的数移给 DM209，DM207 的数移给 DM208，等等。然后把“当前时分”存入 DM200。

提示：OMRON PLC BCD 减运算，如被减数小于减数时，进位位置1（借位），“这个差”为10000 +“被减数”与“减数”之差。要将其变为“差的绝对值”，必须再清进位位，使“0”被“这个差”减。即：10000 –｛（10000 +“被减数” –“减数”｝，即“减数” –“被减数”。但 OMRON PLC BCD 加、减运算时，其进位位也参加运算，这里在未清进位位，故这里使“1”被“这个差”减。

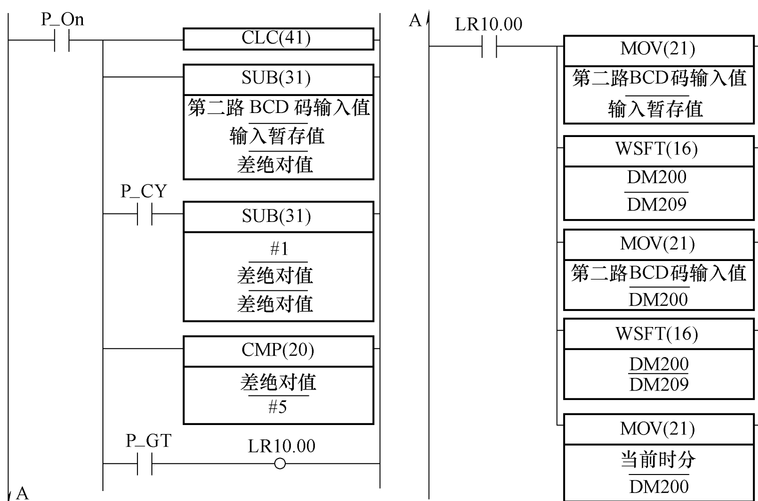


图 6-33　变化采集

可知，这里 DM200 到 DM209 中存的为 5 组数。分别为记录当时的“时分”及与这个“时分”相应的被采集数据。只要变化绝对值超过常数 5，数据就会更新一次。

6.3.3　脉冲量采集

脉冲量，其取值总是不断地在 0 和 1 之间交替变化着的逻辑量。每秒钟脉冲量交替变化的次数称频率。随着脉冲频率的不同，采集的方法也不同。

1. 较低频率脉冲量采集

如图 6-2 所示的电量采集，其频率不高，每秒不到 10 次。用普通的输入点即可进行采集而不丢脉冲。

实现的办法用计数指令。OMRON PLC 无增计数指令，可用可逆计数器指令代替。也可用 INC（加一指令）。但用时一定要令其微分执行。否则，在脉冲的正半周，每扫描周期都将加一。

东电电量采集程序用的为可逆计数器。

2. 较高频率脉冲量采集

如脉冲频率大于 10p/s，但还不到千 p/s。普通输入点不行，可用定时中断或外中断方法采集。

如沈阳华润啤酒厂，用 C200H 型 PLC 采集其生产线装酒的瓶数，最高产量时，每秒要过 10 ~ 20 瓶。为确保脉冲不丢，采取定时中断的办法，执行采集子程序。定时中断时间间隔设为 10ms，即每隔 10 ms，执行一次采集程序。而子程序，就是执行可逆计数器指令。输入一个脉冲，计数器加一。下班时，计数器值转存到存储区并清零。

再如，阜新液压件厂油泵试验系统，在油泵转轴上装有脉冲生成器。油泵转动，脉冲生成器发送脉冲。转速高，脉冲频率也高。故用脉冲频率反映油泵转速。

由于高速时，这个频率较高，故用外中断加定时中断来处理这个脉冲。外中断用的 CPM 机 0. 03。它可设为外中断工作，响应速度可达到毫秒级。外中断的功能是，一旦该输

入点 ON，即调相应的子程序。它还用了定时中断。每 1 秒定时中断一次，调一次中断子程序。图 6-34 所示为此梯形图程序。图 6-34a 为初始化程序。这里“P_First…”是 OMRON 的特殊继电器，仅在扫描第一周期 ON，其它周期均 OFF。用它作外中断及定时中断初始化设定。

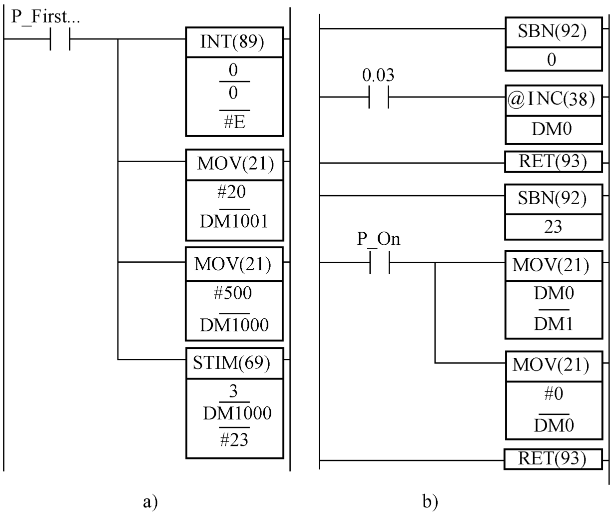


图 6-34 脉冲频率采集程序

a）初始化程序 b）子程序

从图知，INT 指令有 3 个操作数。第一个操作数是 0，含义是允许输入中断；第二个操作数是 0，缺省值；第三个操作数是#E，含义是输入点 3 用作外中断（此外，还应把 DM6628 设为 0001。这也可用 CXP 软件在设定窗口上设），所调的中断子程序号是 0。子程序 0 如图 6-34b 所示。

STIM 指令用以作定时中断设定。第一个操作数是 3，含义是间隔定时中断开始执行；第二个操作数是 DM1000，是低字地址，还有高字地址 DM1001。从图知，在执行 STIM 指令之前，已对 DM1000、DM1001 赋值，一个 20，一个 500，相乘为 10000，含义是定时间隔时间为 1 秒；第三个操作数是#23，指定调中断子程序号是 23。子程序 23 子程序 0 也如图6-34b所示。

执行图 6-34 程序的结果是，只要 0.03 点有脉冲信号输入，系统将调子程序 0，使 DM0 加一，计脉冲。而每经历了一秒钟，系统将调子程序 23，使 DM0 中的数传给 DM1，DM0 清零。显然，这里 DM1 中存的数即为每秒接收的脉冲数，即脉冲频率。

3. 更高频率或三相脉冲量采集

用 PLC 的高速计数功能或高速计数模块。这在本书第 5 章已有介绍。这里不再重复。

6.3.4 脉冲选通采集

DUT-3000 是大连理工大学信息工程研究所研制的 8 路温度采集、控制及传送装置。它的温度数据可用串行接口通过通信的方法传给 PLC。也可用脉冲选通的方法，通过 4 个数据

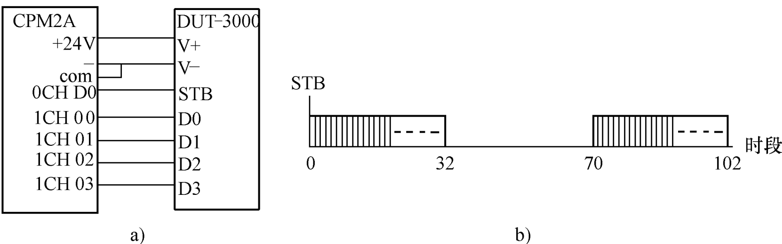


图 6-35 系统布局与信号传送过程

a）系统布局 b）信号传送过程

输入点及一个脉冲选通点，按数位（DIGIT）逐一送给 PLC。DUT-3000 有 8 组温度数据，每组 4 个数位，共 32 个数位。以下介绍后一种方法。系统布局如图 6-35a 所示。

从图知，CPM2A 的 1 通道的 00 ~ 03（4 个 bit）用作数位输入点。选通信号（STB）输入点用 0 通道的 00 点。

DUT-3000 工作时，STB 每隔若干毫秒发一次选通脉冲。每发一次脉冲即依次把 32 个数位逐个通过它的 D0 ~ D3 口，送给 CPM2A 的 1. 00 到 1. 03 输入端。而每发送完这 32 位，停止 38 个时段。之后，又开始新的过程，如图 6-35b 所示。

为了采集这个数据，得有相应的 PLC 程序。图 6-36 为初始化程序。目的是设定外中断、定时中断及工作参数初始赋值。外中断用于选通脉冲输入点，只要出现选通脉冲，即调子程序 0。定时中断设定时，每隔一定时间调子程序 23。工作参数初始赋值是把 0 赋值给 DM110 及把 2 赋值给 DM112。

图 6-37 所示为数据采集子程序。只要出现选通脉冲信号，则采集一个数位。

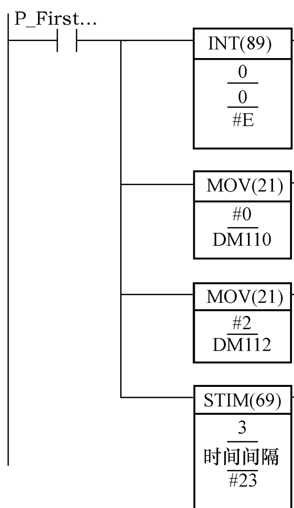


图 6-36　初始化程序

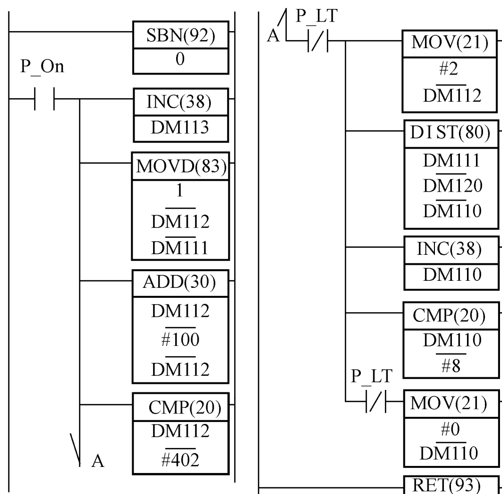


图 6-37　数据采集子程序

从图知，执行一次本子程序，先是 DM113 加 1。再是，第 1 次执行时，把 1 通道的 0 数位，即含有 00 ~ 03 的值，送 DM111 的 0 数位（因此时 DM112 为 0）。接着把 DM112 加 100。再接着进行比较，看 DM112 是否不小于 402。这时是小，故跳出子程序。

第二次执行，把 1 通道的 0 数位，即含有 00 ~ 03 的值，送 DM111 的数位 1（因此时 DM112 为 102）。接着把 DM112 加 100。再接着进行比较，看 DM112 是否不小于 402。这时还是小，故跳出子程序。

第三次执行，把 1 通道的 0 数位，即含有 00 ~ 03 的值，送 DM111 的数位 2（因此时 DM112 为 202）。接着把 DM112 加 100。再接着进行比较，看 DM112 是否不小于 402。这时依然是小，故跳出子程序。

第四次执行，把 1 通道的 0 数位，即含有 00 ~ 03 的值，送 DM111 的数位 3（因此时 DM112 为 302）。接着把 DM112 加 100。再接着进行比较，看 DM112 是否不小于 402。这时不小，故子程序往下执行。注意，这时 DM111 已把 4 数位的数据全部采集到了。

第 1 次执行这后一段子程序时，先是 DM112 恢复为初始值，接着把 DM111 的值传给 DM120（这里用了偏移传送，第 1 次执行，DM110 值为 0）。传后，DM110 加 1。再比较，看 DM110 是否不小于 8，这时不小。则退出子程序。

接着，由于 DM112 已是初始值，故又重复上述四次执行过程。当 DM112 不小于 402 时，第 2 次执行这后一段子程序时，又先是 DM112 恢复为初始值，接着把 DM111 的值传给 DM121（这时 DM110 值为 1）。传后 DM110 加 1。再比较，看 DM110 是否不小于 8，这时还是不小。则退出子程序。

…

直到第 8 次执行这后一段子程序时，先是 DM112 恢复为初始值，接着把 DM111 的值传给 DM127（这里用了偏移传送，第 1 次执行，DM110 值为 7）。传后，DM110 加 1。再比较，看 DM110 是否小于 8，这时不小，把 0 赋值给 DM110，使其恢复为初始化值。子程序数据全部复原。又可进行新一轮的采集。

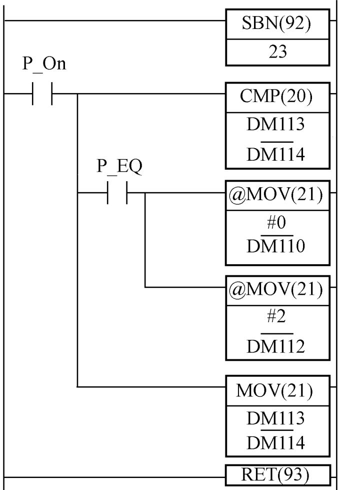


图 6-38　同步处理程序

图 6-38 所示为同步处理程序。从上介绍知，这里的时序关系很重要。一旦时序出错，所有数据将“张冠李戴”，不能使用。为避免出现此情况，特用了同步处理程序。

从图知，它是定时中断子程序。每隔一定时间执行一次。执行时先比较 DM113 及 DM114，看是否相等？相等，则使 DM110、DM112 初始化。不等，则把 DM113 的值传给 DM114。从图 6-36 信号传送过程知道，这里数据传送是有停顿的。目的之一也是为了同步处理。从图 6-38 知，每次调子程序 0 时，DM113 的值总是加 1，是变化的。而不调时，即传送停顿时，它的值不变。此程序正好利用这个不变，使使 DM110、DM112 初始化。显然，DM110、DM112 初始化正确，也就确保了这个同步了。

6.3.5　采集数据暂存

有时不希望采集到新数据后，立即把旧数据清除，而是要暂存若干采集周期。这样可保证即使上位机未及时读取，所采集的数据也不会丢失。图 6-39 所示为处理这类问题的一个程序。它用于运动控制时坐标值的采集。这里的暂存区为“x 区开始地址”到“x 区结束地址”。

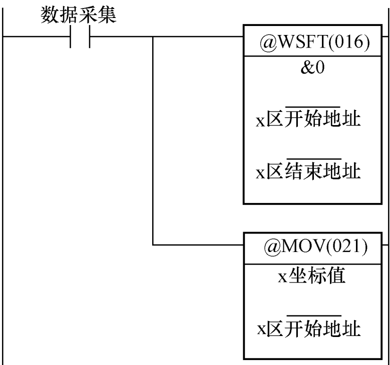


图 6-39　数据暂存程序

从图知，当每次“数据采集”信号 ON，将执行一次 WSFT 及 MOV 指令。前者把“x 区”的各个字依次作字移位。原“x 区结束地址”的值丢失，而被它的前一个地址的值取代。“x 区开始地址”的值变为 0。后者把“x 坐标值”（新采集的值）赋值给“x 区开始地址”。这样，从“x 区开始地址”到“x 区结束地址”，存储的都是“x 坐标值”，只是采集的时刻依次

不同而已。

6.4　数据录入程序设计

关键词：数据录入、用通用指令录入、用特殊指令录入、编码键盘录入、数据模拟方法录入

数据采集主要通过 PLC 的输入点、输入通道或通信口实现。

6.4.1　录入数据设备

PLC 录入数据的设备有 PLC 厂家提供的设备，如简易编程器、简易数据设定器及人机界面。还有用户自己配备的设备，如简易数字键盘。

用简易编程器、简易数据设定器录入数据，按有关说明操作就可以了。人机界面的使用将在本书第 7 章讨论。

此外，在小型 PLC 面板上，多还有两个电位器，用它，也可粗略地实现向 PLC 录入数据。

简易数字键盘有 0～9 数码键或再加 A～F 字母键，或再加一些如“确认”、“清除”等操作键，也比较简单。它可用于在现场，向 PLC 录入数据。如前已介绍的辽源汽车油泵厂油泵测试数据记录，就使用这种键盘。

由于 PLC 厂家不提供这种键盘，所以目前这个输入只好用普通的输入点。这当然不太经济。不过，OMRON 公司还总算理解到用户的需要，在有的 PLC 上，开发有键盘录入指令。用了这些指令，可减少使用输入点。经济上较合算。

此外，也可用拨码开关录入数据。每个开关可任意置 0～9，或 0～F 多个值。正好一个数位（digit）。而每个开关按 8421 编码，与 PLC 的 4 个输入点相接。一个输入通道，4 个数位，接 4 个拨码开关，对应一个字。一些定时器、计数器的设定值，要在现场确定时，可使用这样的输入通道实现。

再者就是一些智能设备，如条码读入器，可利用 PLC 串行接口或外设口，向 PLC 送入数据。这是有关通信、联网的问题，在此就不讨论了。

6.4.2　用通用指令录入

多数 PLC 录入数据用通用指令。图 6-40 所示为用数位（digit）移位（SLD）及数位传送（MOVD）指令，实现向目标地址录入数据的梯形图程序。

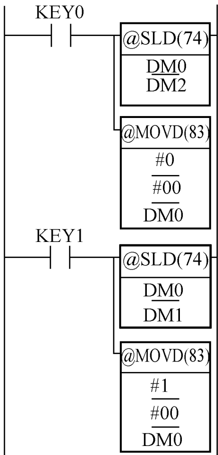


图 6-40　键入数位 1

该图只画出键 0 及键 1 的情况。当任一键按下，先是，使 DM0、DM1（可看成双字长的数，即 8 位数）中的各数位的数从低到高移位。然后，把此键值，如“0”、“1”、“2”…送入最低位。注意，这里的指令为微分执行是必要的。

可知，要送入完整的数据，必须键入 8 次。

图 6-41 所示为用使用 BCNT、SLD、DMPX 等指令，实现向目标地址（目标低字到目标高字间）录入数据的梯形图程序。

该图用的是字处理指令，程序很简洁。当任一键（这里只定义

10 个键 0 ~ 9，对应输入点输入通道的第 0 ~ 9 位，也可增多）按下，则 200 通道大于 0，这将使目标低字到目标高字移位，然后，把此键的值（见 DMPX 指令的含义），如“0”、“1”、“2”…送入最低位。注意，这里的指令也应为微分执行。

图 6-42 所示为目标地址可选的录入程序。

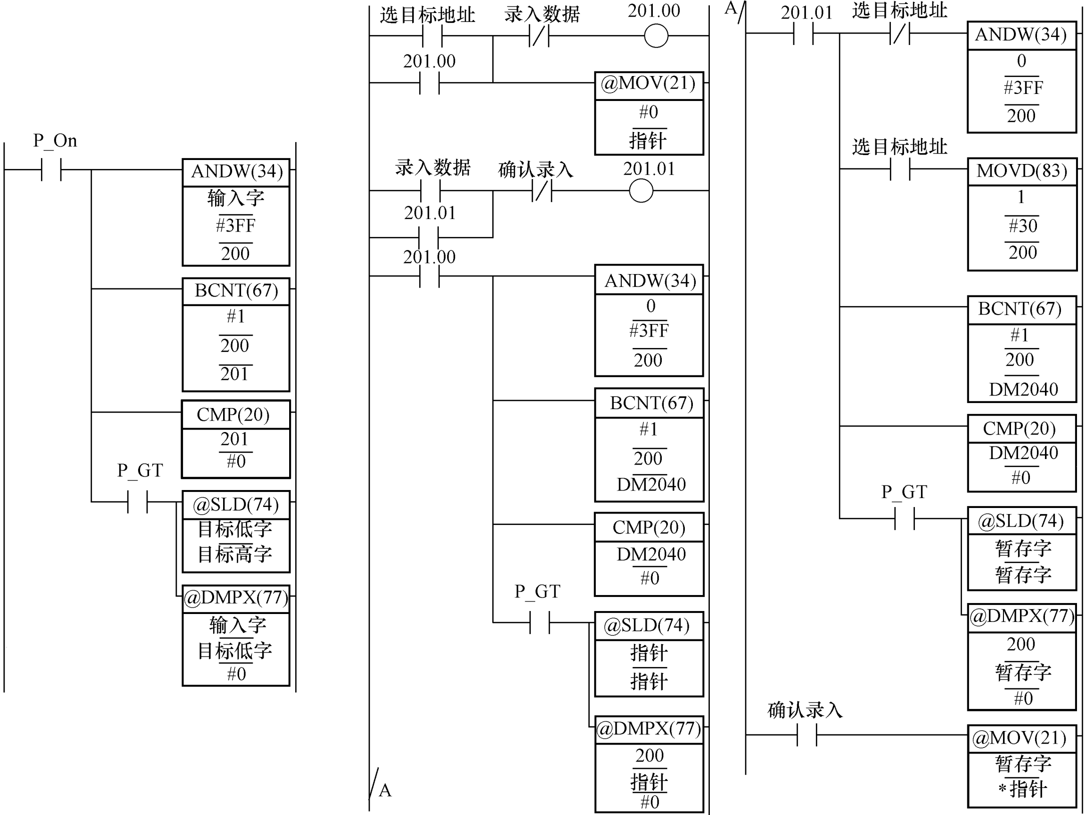


图 6-41　键入数位 2　　　　图 6-42　目标地址可选的录入

该图用的指令与图 6-41 相同。只是先选定目标地址（对指针赋值），后录入数据（向指针指向地址送数）。具体过程是，先使“选目标地址”ON，这时把“指针清 0，同时，201.00 ON，这时，录入数据即为目标地址，即向指针赋值。地址送入后，再使“录入数据”ON（这时，“选目标地址”应 OFF），则 201.00 OFF，201.01ON。这时，录入数据将送“暂存器”。最后，使“确认录入”ON（其它的均 OFF），则使 201.01 OFF，停止录入，同时把“暂存器的内容送指针指向的地址。

可知，这个程序更灵活些，可向 DM 区任何地址送数。

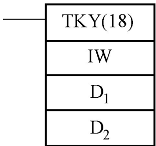
6.4.3　用特殊指令录入

OMRON C200HS，CQM1，CQM1H 和 C200HX 型 PLC 有 TKY（10 键）及 HKY（16 键）指令，可用于录入数据。

1. TKY

其功能是，通过 10 个输入点，用 0 ~ 9 十个数码键，实现向指定字录入数据。

其梯形图格式为



这里　IW——指定输入通道，用其中的 00～09 位；

D_1——指定 D 及 D1 +1 组成两字的目标移位寄存器；

D_2——指定另一目标字。

硬件接线应做到，键的值与输入点号对应。如 3 键，则应接到 03 点。

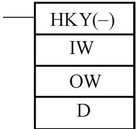
执行本指令，当这里 0～9 任一键按下时，将使目标移位寄存器所有数位左移，新键入的数字（键值）存入最低位，原最高位数丢失。同时，D2 的相应位 ON。如键 2 按下，则 D2 的 02 位 ON。

可知，本指令与图 6-40 及图 6-41 的功能是相同的。

2. HKY

其功能是，通过 4 个输入点及 4 个输出点，用 0～F 16 个数码键，实现向指定字录入数据。

其梯形图格式为：



这里　IW——输入通道，用其 00～03 四个输入点；

OW——输出通道，用其 00～03 四个输入点；

D——D 及 D +1 组成两字的目标移位寄存器，D +2 为另一目标字。

硬件，应按说明书要求接线。

执行本指令，当这里 0～F 任一键按下时，将使目标移位寄存器所有数位左移，新键入的数字（键值）存入最低位，原最高位数丢失。同时，D +2 的相应位 ON。如键 2 按下，则 D2 的 02 位 ON。用此指令也只需用 4 个输入点。

此外，还有 DSW（数码开关键入）、MTR（矩阵输入）指令。可用于从数码开关录入更多的数据。

6.4.4　用编码盘录入

如有如图 6-43 所示编码键盘，有 0、1…，共 15 个键。但自身有硬件编码器。根据不同键按下（要做到，只能一键按下有效，多键按下无效），在其输出 8421 端，将有按 2 进制编码的不同的通路。如按下 7 键，则 4、2、1 三端各与 COM 端通，8 端不通，代表 2 进制 0111。其它 1 到 9 键，与此类似。0 键按下，4 端全通，即 1111，用以代表输入 0（内部程序可做处理，以实现的这个目的）。还有 ABCDE

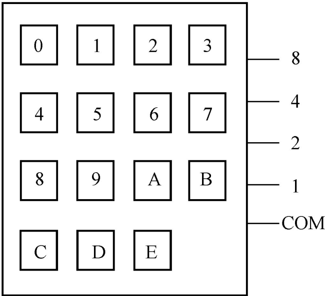


图 6-43　编码键盘

键，分别用 1010 、1011、1100、1101 及 1110 编码，可用作操作键，如清除，退格等。

有了以上硬件条件，怎么通过程序实现数据录入？图 6-44 所示为一个实际程序。

在图中，先取“数据输入通道”的最低数位存于 200 通道中（假设 8421 分别接此通道的 03、01 点，若不是应另作处理）。然后把 200 通道与常数 0 比较。如大，说明，200 通道有大于 0 的数据，置“有数据输入”ON。

当“有数据输入”ON，先看此数是否为 FFFF，如是令其变成 0。因为，前已假设用 FFFF 替代 0。如不是，不做处理。

最后，目标字的各数位各向其高一位移位及把输入的数送目标低字的最低位。此即完成了一个数位的录入。与本节图 6-40 相同，如果目标字是两个字，则要录入 8 次，才能完成这个数据录入。

显然，有了这个编码键盘，用 4 个输入点就可进行 15 个数码的录入，是较合算的。

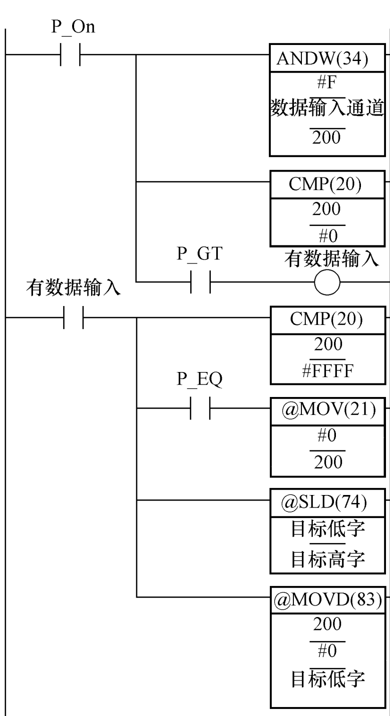


图 6-44 编码输入程序

6.4.5 用模拟方法录入

以上介绍的都是以数字形式录入数据。其实，PLC 还可用模拟形式（方法）录入数据。有两种方法：用电位器；用定时示教录入。

1. 用电位器

如 CPM 型，S7-200 型 PLC，其面板上都有两个电位器，它直接与 PLC 指定通道关联，如 CPM2A 的两个电位器分别与 250、251 关联。当电位器旋钮顺时针转到头时，通道值为 200；当电位器旋钮逆时针转到头时，通道值为 0；中间位置时，按比例居于 0～200 之间。显然，用户可利用电位器旋钮处于不同位置，使 PLC 得到不同的输入值。

遗憾的是，这样输入没有指示，只能凭操作者的感觉。而且，精度也不高。故只能用于要求不高的场合。

2. 用定时示教录入

除了用电位器，还可用定时示教录入。图 6-45 所示即为这个梯形图程序。

从图知，它用于定时器 007 的定时值（即图中的“设定值”）的设定（录入）。当“设定开始”（为 PLC 的某输入点）ON，可逆计数器 080 开始计数，每 0.1 秒增 1。如连续按几秒，则这计数器的值为几十，为按得的时间（以秒计）的 10 倍。一旦“设定开始”OFF，则先是把可逆计数器 080 的计数值送“设定值”，进而使计数器清零，为下一个设定作

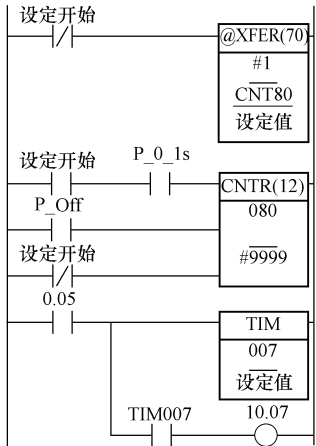


图 6-45 定时示教录入

准备。

提示：图6-45 XFER 传送指令为微分执行是很必要的。否则“设定值”将永远为0。

显然，有了这个程序，定时器 007 的定时值，将是“设定开始”ON 的时间的 10 倍。这也就是说，其工作延时也就等于“设定值”ON 的时间，即这个操作的示教结果。

6.5　数据存储程序设计

关键词：记录、定时存储、事件存储、固定地址存储、多对象存储、压缩存储、安全存〖、读写保护、数据加密

数据采集也是存储，是把 I/O 通道的数据存到指定的内存地址中。但它的存储区小，只是数据暂存。有了新数据，旧的将被取代。而数据存储则是新旧数据依次存储。有了新数据，旧的不被取代，而是与旧的同在。当然，这个“同在”是有限制的，与存储区的大小及每次存储的数据量有关。存储区大，数据量小，则“同在”的新旧数就多。反之，就少。

而且，为了便于使用所存储的数据，一般在进行存储前，先要按一定格式，把数据组织成记录，再按一个个记录，连续地存储在一个数据区中。

此外，为了节省存储空间，也可压缩存储。为了安全，也可加密存储。

对这些问题将在本节逐一进行讨论。

6.5.1　记录存储

1. 记录

是指一组有关联的数据。记录要有关键字（或多字），以区分不同记录。

如以上讲的（图 6-2）电量采集数据，所存的记录就是一组数，共 4 个字。第 1 个字存年月。第 2 个字存日及时段。第 3、4 个字存电量累计值（用双字长）。每天分 7 个时段定时存储，每天 7 个记录。

再如以上讲的（图 6-4）硫化机合模、开模，所存的记录就是一组数，共 4 个字。第 1 个字，存对应的硫化机号（3 位数）及开（用 A 表示）或合（用 B 表示）模标志。第 2 个字，存年月。第 3 个字，存日时。第 4 个字，存分秒。只要开或合模一次，即被监测的数据有了变化，则存储一个记录。但由于硫化机工作时间是有规律的，大体每小时，开合模一次。所以，它每天的存储长度可预计。

再如还是以上讲的（图 6-4）硫化机的温度监测，也是按记录存储。每记录为 5 个字。第 1 个字存对应的硫化机号。第 2 个字存年月。第 3 个字存日时。第 4 个字存分秒。第 5 个字存储当时的温度值。它也是变化存储。但只是出现超温时才存储。但什么时候超温则是随机的，所以，它每天的存储长度不好预计。

数据存储的记录除了定长的（固定格式），也有非定长的（非固定格式）。可按实际情况组织。后者，可节省存储空间，但程序的算法要复杂些。

组织成记录后怎么存储？存储区的地址怎么分配？方法也很多。可以是地址不固定，当数据存储时，按数据区地址升幂（或降幂）顺序依此存储。当数据满后，又回到起点，用新的记录取代旧的记录继续存储。也可以是固定地址的，什么时候、存储在什么地址是固定的。但多数用的是地址不固定的。

什么时候存储可按时间设定，每天有固定的存储时间，这叫定时存储。也可为事件引发，当发生所定义的事件时，才存储，这叫事件存储。

数据存储方法一般用间接地址，即指针访问。这样的程序较简练。也可用在数据采集程序中用过的字移位，如图6-32、图6-33 所示。只是存储区大，执行这样的移位指令，执行时间可能很长。

如有不同对象的数据记录，可一个对象存储在一个存储区。也可多对象混合存储在一个存储区。

2. 定时存储

分时段记下监测量的当时值，有时还要记下当时的时间。图6-46 所示为定时存储例子。

从图知，它先把“当前时分”与“时间设定”进行表比较。“当前时分”是从 PLC 时钟中读出，随时间而变。“时间设定”是一组数，按要存储的时段划分。如700 代表7 点0 分。通道“时段”中的相应位是 ON，还是 OFF，与比较的结果有关。如“时间设定”的第一个数为700，而“当前时分“又正好7 点0 分，则“时段”通道的“时段1”位 ON。等等。

当这“时段1”等 ON 时，将把“当前日时”等数据存入指针指向的地址。存一个数，修改一次指针（地址加1）。可见，它是定时，但变地址存储。

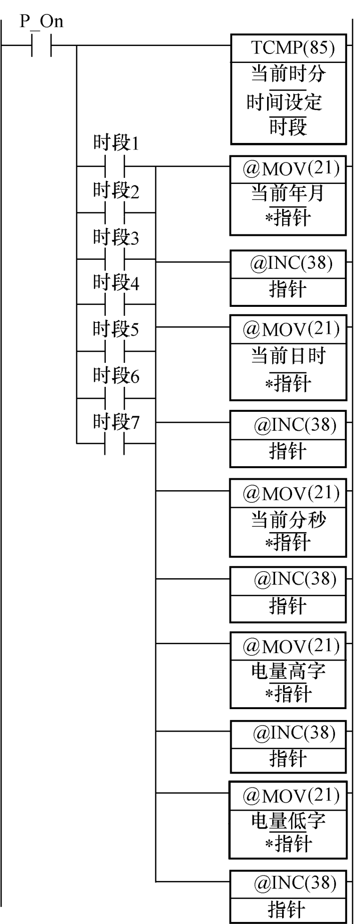


图6-46 定时变地址存储

图6-47 所示为指针控制程序。

从图知，它始终进行“指针”与“存区起始地址”及“存区结束地址”比较。只要指针不在此区间，则用“存区起始地址”赋值给“指针”，使指针复原。对指针的这个控制，可确保数据始终在存储区中，周而复始地存储。

3. 事件存储

发生某个事件，如监测量超限、监测量变化（对开关量），或变化超过某范围（对模拟量），就进行存储。存储时，不仅记下监测量的当时值，还要记下当时的时间。

图6-48 所示为超限存储梯形图程序。但该图未把全部程序画出。

该程序用于图6-4 介绍的轮胎硫化机温度超限纪录。从图知，如“一号合模”ON，即一号机工作，则不断的进行“温度值”与“温度上限”及“温度下限”比较。一旦超限，即这里的 LR14.00 或 LR14.01ON，则都会进行数据存储操作。这时，先是把“当前年月”存入“指针”指向的地址，后修改指针。接着存“当前日时”… 以下存储在图中未画出。当然，如图6-47 所示的指针控制程序也是绝对需要的。这里也未画出。

4. 固定地址存储

以上两例，都是非固定地址存储。特点是“指针”随存储修改，存储区可很大。

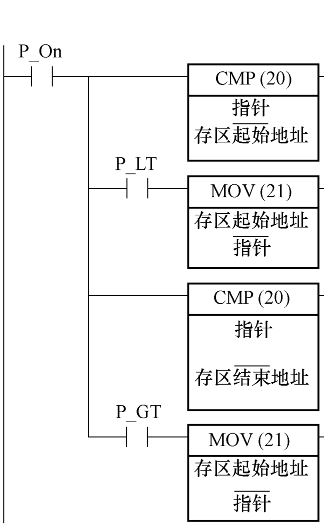


图 6-47　指针控制程序

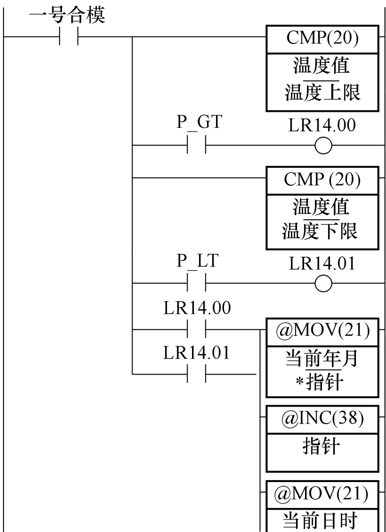


图 6-48　超限存储

固定地址存储。特点是“指针”用存储事件赋值。发生什么事件，就有什么事件的“指针”值，因而，该事件的数据存的地址也就固定了。图 6-49 所示为固定地址存储的梯形图程序。该程序是每“整 5 分”存储一次数据。存储地址与存储的时刻有关，是固定的。

从图知，它先把当前的分除以 5。其商存于 HR0 中，余数存于 HR1 中。接着，判断 HR1 是否等于 0。如相等，即为“整 5 分”，则 LR11.00 ON。进而，起动数据存储。但在存储数据前，先根据“当前时分”计算指针值，然后，再按“指针”指向的地址存储“存储数据”。所有的计算与存储指令，都是微分执行的。目的是确保是在进入“整 5 分”0s 时，存数据。

从程序计算情况可知，当时间从 0h 0min 到 23h 55min 之间变化时，“指针”的值将在 0 ~ 287 之间变化。即“存储数据”将存储在 DM0 ~ DM287 之间的数据区中，而且，不同的存储时间，有自身的固定 DM 地址。

固定地址存储时，时间值可不存。因从地址，即可知道它是什么时间的“存储数据”。也很节省存储空间。

5. 多对象存储

非固定地址存储，可存储多个对象。如图 6-4 所示轮胎硫化机实时纪录，用的就是这种存储。它把各个轮胎硫化机的数据，都存储在一个 4K 的存储区中。用统一的存储指针存储。但不同的轮胎硫化机，有不同的关

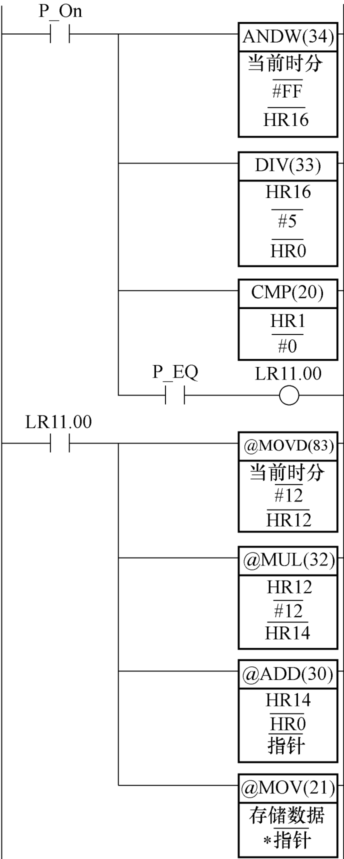


图 6-49　固定地址存储

键字，用关键字区分它们的记录。

多对象存储把存储区连成一片，较节省存储空间。而且上位机读此数据时也简单，发一个读命令，等着接收与应答即可。上位机读取数据后，可按标志的不同，把数据分开，并进行处理及存储。

6.5.2 压缩存储

为了节省数据的存储空间，可对数据编码长度进行压缩。数据压缩后，再行存储，即压缩存储。读取压缩存储的数据要解压缩，才能清楚它的含义。一般讲，这是上位计算机要作的事。只要清楚数据是怎么压缩的，解压缩是不难的。

压缩存储不仅节省存储空间，而且，也节省数传输的开销。

数据压缩分为两种：一种和语义无关，压缩时只从数据的形式出发，不涉及数据信息内容，这种技术适用于任何数据。另一种与语义相关，压缩时考虑数据信息内容和语义，去掉其中一些冗余的信息。计算机常用的压缩技术有模式替代、压缩重复字符、免写空字域、编码替代、不等长编码、相邻数据利用、消除冗余信息等。

对 PLC，最基本数据压缩技术有：

1. 用十六进制码取代 BCD 码

BCD 码中每一位都用 4 位二进制数存放，表达的数字仅是 0～9 这 10 个数码。其实，4 位二进制数可表达 16 个数码。所以，用 BCD 码存数存在资源浪费。一个字（16 位二进制数），用以表达（或存）BCD 码数，最大只能到 9999。而用以表达（或存）16 位二进制数，最大可达 65535。如果预先将它转换成二进制就能节省很多存储空间，而且还便于算术运算。

2. 合并数据代替单独数据

日期数据，有年、月、日，要用两个字，4 个字节表示。如 2004 年 6 月 30 日，则写成 2004（一个字），06（一个字节），30（一个字节）。当然，如千年不考虑，用 3 个字节也可，如上例，用 04，06，30 也可表示。如把这数据合并，合并后再存储，就不要 3 个字节了。

合并数据的方法是，各合并数先加权，后相加。如本例月的先乘 40（一个月最多 31 天，为了好算），年先乘 400（年最多 366 天、考虑闰年，为了好算），日不乘。这样，2004，06，30 的日期，合成的值将是 1690。如果计算到 2099 年 12 月 31 日，则此值为 44431。用 16 进制格式存储这个数，两个字节就够了。存储长度压缩了 1/3。

数据还原也不难，如 1690，先被 400 整除，得 4，即 2004 年。再把它的余数，270，被 40 整除，得 6，即 6 月。而后者的余数，30，即 30 日。

处理时、分、秒，也可用类似方法。但为了充分利用内存，最好把年（只考虑 80 年）、日、时，分在一组，用一个字存，而月、分、秒，分在一组，也用一个字存。两个字，即可把本应用 3 个字才能存的年、月、日、时、分、秒，进行存储。

3. 存相对值而不存绝对值

对数值数据可以采取求差法，例如电量，存增量，一个字节就够了。需得知月累计值，可在读出数据后计算。

但这么做，数据不太安全。万一有一次存储数据丢失，累计值就不对了。故一般不用。

4. 以位（bit）计算存储单位

每个记录不以字或字节为单位，而二进制位（bit）为单位，计算存储长度。这可充分利用剩余的二进制位，提高存储效率。只是算法复杂些。

5. 高级数据压缩技术

原则上说，数据压缩技术与数据的语义有关。合理的压缩是去掉数据中的无关信息，而保存有关信息。数据的形式虽有变化，但它所保存的有关语义信息没有变化。具体的方法很多。在此不多介绍了。

6. 问题

数据压缩也带来一些问题。数据压缩会增加程序系统的复杂性；被压缩后的数据可能会失掉原有的良好的标准形式，降低它的通用性、可移植性及可靠性。因此，是否采用压缩技术需要根据具体问题来决定。

6.5.3　安全存储

1. 数据安全

数据安全是指，数据不能丢失，即使丢失，最好也能找回来；不能被没有授权的人读写，即使把数据读取了，也令其无法使用。

对 PLC，数据不能丢失是不难的。因为它的数据存储区都是或都可是掉电保护的。即使停电了，其所存的数据也不会丢失。再就是，在程序上要使用好地址及控制好指针，要确保存储区的数据不被误改写。

要是能把丢失的数据找回来，就要作数据备份。备份是原数据的拷贝。有了新型的内存卡，PLC 也可作数据备份了。这也是一种重要的安全机制。

2. 读写保护

为了数据安全，作好写保护是需要的。OMRON PLC 进入运行方式，其数据只能由指令写，人工或通信都不能写。这也是很好的数据保护。

有的 PLC 为防止在联网中，或外设操作中，数据丢失，或不宜被别人访问，设有相应的管理指令或设定。执行了这类指令，或作了这些设定，数据将不会被读写。可确保数据安全。

3. 数据加密

如果读写保护不好实现，也可对数据加密，这样，即使数据被人读取了，但如果不知是怎么加密的，则无法理解其含义，也可保证数据安全。

加密是对数据进行混乱的拼凑或者隐藏信息的过程，从而使数据难以理解，除非进行解密或者破译，把它转换成原始的样子。加密是防止非法使用数据的最后一道防线。

在使用这些数据和文件之前再将其还原成原来样子，称为解密。密码数据被第三者截获，探知其真实内容，称为破译。

其实，数据压缩后，如不知加压的算法，也无法解压。这样的数据也等于是加密的数据。被读走也无妨。

数据加密的方法通常是用一个通用的算法再加上一个密钥。算法可以是公开的，但密钥是保密的。加密过程就是把该算法施行于要加密的数据，密钥作为控制参数控制加密过程。常用的加密方法有三种：移位法、代替法和代数法。

移位法是将原来数据中的字符顺序重新排列。

代替法是用替代字符取代信息中的字符，或用替代字符组来取代信息中的字符组。

这两种方法简单，但不可靠，容易被破译。

代数法是先把信息转换成等效的数字序列，然后用代数方法对数字序列施行变换，得到密码。例如把数字序列和密钥的二进制码作异或运算得到密码。解密过程也很容易，用同一密钥再作一次异或运算就可得到原文。

数据加密，虽能保护数据安全，但也增加不少麻烦。只是非常需要时，才用它。

6.6 数据显示程序设计

关键词：数码管、7段码、动态输出模块、信息显示、脉冲选通显示

作为数据终端，按要求显示数据是非常必须的。没有数据显示，用户就无法与 PLC 对话。录入数据时若没有数据显示，就不知道录入的数据是否正确，当采集数据时，也须知道采集的数据是否正常。等等。

在 PLC 上有很多指示灯，可用以观察 PLC 的 CPU 单元、I/O 单元、特殊单元及通信单元的工作情况。但 PLC 的内部数据，PLC 本身无法显示。

为此，要用 PLC 的外设。如简易编程器、数据访问器、高档及低档人机界面（可编程终端）等，可通过 PLC 的外设口或通信口与 PLC 连接，用以显示与修改 PLC 的内部数据。只是这些设施价格较贵些。而且也稍有不便。

此外，还可用输出点驱动数码管作数据显示，而且数码管还可通过数据脉冲选通的方法显示数据，以节省输出点的使用。

近年来，不少公司，为自身的 PLC，开发有多功能简易显示模块（5DM），不通过通信口，直接与 PLC 连接，可实现数据显示。也有的在 CPU 模块上集成有显示界面。有的还另设有输入按钮，或在画面上设有触摸按钮。一定程度上可实现可编程终端的功能。

以下分别对这些显示进行介绍。

6.6.1 数码管数据显示格式

数码管有 7 段显示灯，如图 6-50 所示，可用以显示 0 ~ 9 间的 10 个数码。

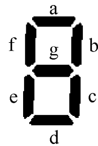


图 6-50 7 段码显示

从图知，这 7 段码不同的显示组合，将反映不同的数值。如这里 a、b、c 3 段亮，其它的不亮，则显示数码 7。再如，仅 b 不亮，其它全亮，则显示数码 6。用这种 7 段的不同组合去显示数字，故称 7 段码。

为了显示 7 段码，数码管要有 7 个与这 7 段码对应的接口及电源端口。而为控制这个显示，PLC 就要用 7 个输出点。

这显然是不经济的。为此，已作了两个改进：

一是，很多数码管已内置有译码电路，可把 8421 码或 BCD 码自动译成 7 段码。8421 码或 BCD 码用 4 个接口加电源接口，就可接收 0 ~ 9 间的 10 个数字信号。PLC 控制这个数字，则也只要用 4 个输出点。

二是，PLC 用动态输出模块（也称多点输出模块）去控制数码管。OMRON 动态输出模

块，价格比普通的贵一倍，但动态输出的点数可达 128 点，为普通的 8 倍。

显然，经此两项改进，一个动态输出模块可控制 32 个数码管（每个管 4 点），可显示 2 个字的信息。

6.6.2　数据动态显示

用动态的方法显示数据可减少输出点，是数据显示的好方法。动态方法可用 OMRON 动态输出模块实现，也可用指令选通方法实现。

1. 动态输出模块实现

用多点输出单元代替普通的输出单元，用以显示数据是较合算的。

一个多点 I/O 单元，动态使用时，控制 128 点，占 8 个通道。但它占用的不是 I/O 通道，而是内部辅助继电器的通道，故称之为特殊单元。

动态输出由数据与选通信号组合生产。它分成两组，每组有 64 个输出点，分别有 8 个选通信号及 8 个数据信号。仅 OD 型多点 I/O 单元，可用于多点输出。使用时，要作相应的设定。

数据输出可设定成正或负逻辑。正逻辑时，数据位 ON，对应的输出端也 ON（高电位）；而负逻辑时，正好相反，数据位 ON，对应的输出端 OFF（低电位）。

选通信号总是负逻辑。出现选通信号，其对应端点 OFF（低电位）。选通信号自动循环地出现，周期 32ms，作用时间 2ms。

图 6-51 所示为动态输出的定时图。

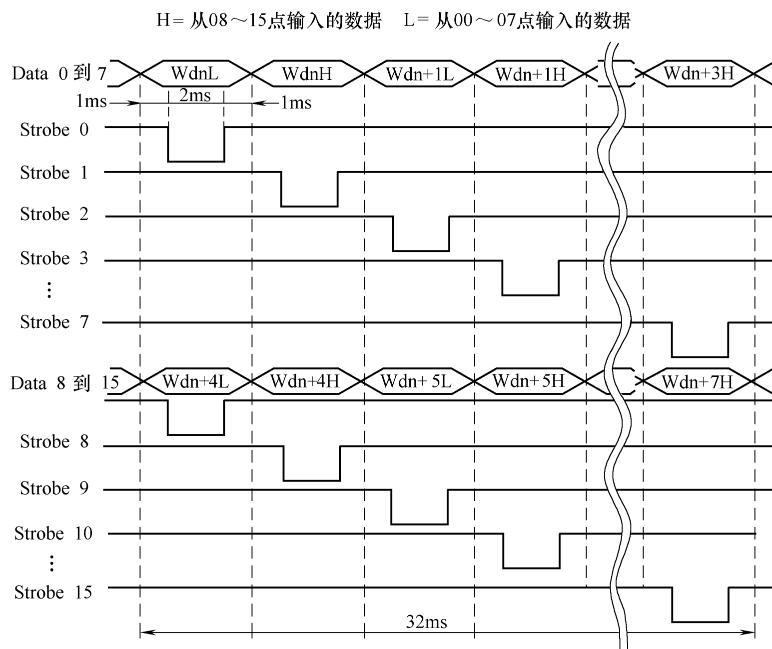


图 6-51　动态输出定时图

从图知，它有 16 个选通信号 Strobe 0 ~ Strobe15，16 个数据信号 Data0 ~ Data15。第一个选通信号 Strobe 0、Strobe 8 激活时（负，2ms），Wdn L 及 Wdn +4L 输出。第二个选通信号

Strobe 1、Strobe 9 激活时（负，2ms），Wdn H 及 Wdn +4H 输出。……这里两组选通信号 0 ~7 与 8 ~15 是并行工作的。

由于是分时工作的，所产生的输出对负载的作用，只是在选通的 2ms 时才存在。要使这个作用保持，数码管要有锁存器，用以锁定这个分时输出的信号。

此外，动态输出为分时的，有延时，最大的延时要 32ms。比普通输出的延时长。

2. 指令选通方法实现

图 6-52 所示为选通显示数码管组件。

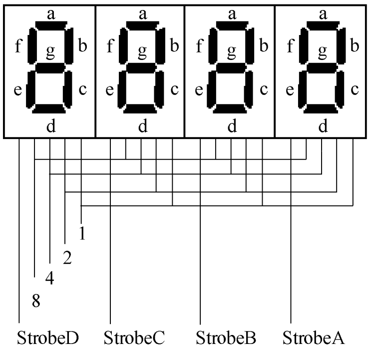


图 6-52　选通显示数码管组件

从图知，这里每个数码管都有 4 个 8421 二进制码输入端，每个管的这 4 个端是又分别相连。8421 端与 PLC 的 4 个半导体的输出点相接（继电器接点速度低，有不适经常通断，故不适于显示数据）。如 1 接 1100，2 接 1101，4 接 1102，8 接 1103。这样，这 4 根输入线组合，即与 11 通道的最低数位的值有关。将在 0 ~9（BCD 码）或 0 ~F（十六进制码）间取值。

图中每个数码管，分别有一个选通信号输入端，如 StrobeA、StrobeB…等。硬件设计成，当选通端有信号（高电平），8421 端的当时数据生效；当选通端无信号（零电平），8421 端的原数据保持。

有此硬件，再加上图 6-53（用于生成选通信号）及图 6-54（用于输出数据）所示程序，用 8 个 PLC 的输出点，4 个用于接 8421 端，4 个用于接 4 个选通端，即可实现一个字的数据显示。

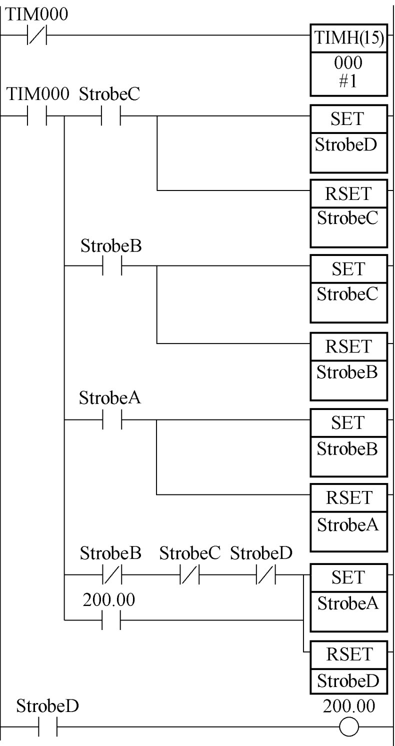


图 6-53　选通信号生成

从图 6-53 知，该程序用了高速定时器 0（TIM 000）。按图中参数选择，它将每 10 毫秒 ON 一个扫描周期。每当 TIM 000 ON，将依此使 StrobeA、

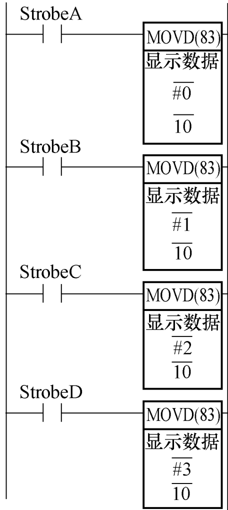


图 6-54　数据输出

Strobe B、StrobeC 及 StrobeD 不停地轮流 ON 10 毫秒。图中用 200. 00 作 StrobeA 过渡，是为了“同步化”。是非此不可的，其道理详见第 3. 4 节。

图 6-54 所示为数据输出程序，它只是依不同的选通信号，从显示数据字（通道）中，选择不同数位显示。4 个选通信号，4 个数码管，正好显示一个通道的数据。只是这里也是动态工作的新数据显示将有延迟。

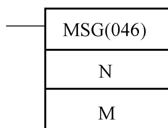
在图 6-53 的基础上，如硬件再作改进，对选通信号编码。4 个选通输出点，可编成 16 个循环的选通信号。那样，4 个 8421 数据输出加 4 个选通输出，就可显示 4 个通道的数据，是较合算的。

用数码管显示数据非常直观、好看，而且可大可小。在 PLC 早期的数据显示手段中，它是很常用的。

提示：总的看，简易键盘加数码显示是较原始的数据录入及数据显示方法。用可编程终端是较好的方法。PLC 基本上不用编程，也不占输入、输出点，即可用以录入和显示数据。

6. 6. 3　简易编程器信息显示

OMRON 有用于在简易编程器上显示信息（ASCII 码）的指令，即 MSG（046），其梯形图指令格式为



这里　N——信息号，必须是十六进制 0000 ~ 0007（或十进制 0 ~ 7）;

M——信息首字。

显示信息时，M 指定包含 ASCII 信息的第一个字的地址。消除信息时，M 可以是任何的十六进制常数（0000 ~ FFFF）。

执行条件为 ON 时，MSG（046）对 N 指定的信息号，注册 M ~ M + 15 中的 16 字的 ASCII 数据（包括空字符在内最多可达 32 字符）。一旦一个信息已经注册，当连接编程器时，该信息将显示在产生的错误信息之后。但在信息注册后，还可通过重写信息存储区的信息来改变信息显示。

要想清除已注册的信息，执行 MSG（046）指令，把 S 设置成将要清除信息的信息号，将 N 设置成常数（0000 ~ FFFF）。

即使程序停止执行，在程序执行时注册的信息，仍然一直保持，但当程序再次执行时，所有的信息都将被清除。

例： 梯形图程序如图 6-55 所示。显示数据如图 6-56 所示。

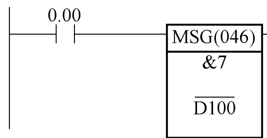


图 6-55　显示程序

当 0. 00 ON，D100 ~ D115 中的 16 个字转换成 32 个 ASCII 字符，并作为信息号 7，显示在简易编程器上。

有了以上程序，还得有简易编程器，才能有以上的信息显示。只是在目前，简易编程器已用的很少，故用此法显示信息已不多用。

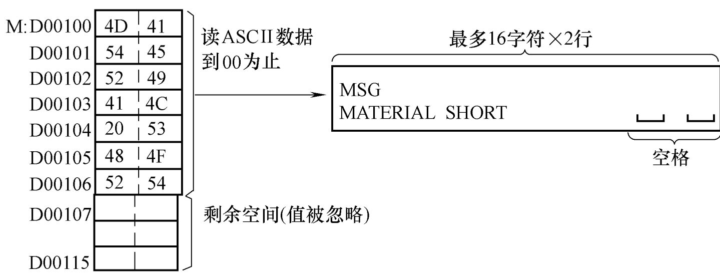


图 6-56　显示数据

6.6.4　数据脉冲选通显示

它用了前面提到的数码管脉冲选通数据显示器。字数据用了脉冲选通的方法（使用一个输出点），按位（bit）逐一传送（也使用一个输出点）。其过程靠 PLC 程序实现。而数据到了显示器后，转换成 7 段码及进行显示，则是该显示器自己能实现的功能。

图 6-57 所示为一个实现此功能的 PLC 程序。

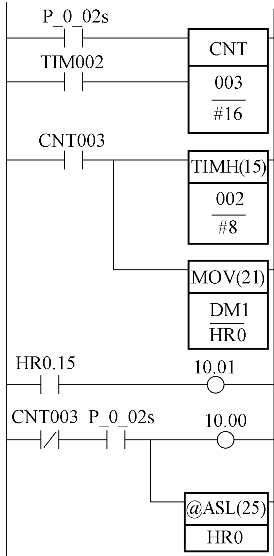


图 6-57　数据脉冲选通显示程序

该图用了一个计数器 CNT003，进行 16 ~ 0 减计数。计数用脉冲是 P_0_02s，为 20ms 脉冲。从 16 减到 0 后，计数停止，CNT003 ON 。而 CNT003 ON 将使定时器 TIMH2 工作。经 80ms 延时，计数器复位。计数器复位又可计数，又重复上述过程。为了等待处理及工作同步，这 80ms 暂停传送是必须的。

要显示的数据存于 DM1 中。当 CNT003 ON 时，用 MOV 指令，把它传送给 HR0 通道。当 CNT003 OFF 期间，用移位指令逐位传给 10.01，而选通脉冲用 P_0_02s 控制 10.00 产生。这两者配合，即可把 DM1 的 16 位（BIT）的值，逐一传给该数据显示器。

显然，这里的数据显示是有延时的。每隔 20ms 传一位（BIT）数，16 位需 320ms，再加等待 80ms。这里未计及I/O 刷新，最少需经 400ms 才能完成一个字的显示。不过不到半秒的延时，问题是不大的。

这里只用一个字移位，实现一个字的显示。其实也可用两个字，以至于更多字的移位，以实现两个字，以至于更多字的显示。只要显示时间延长能够忍受，理论上讲再多的显示都是可以的。这样以时间的延长换取空间的节省是较合算的。

还应指出，从本质上讲，图 6-57 的方法与 6. 6. 2 介绍的用动态的方法显示数据的方法是相同的。只是使用的指令不完全相同。

6.6.5　高档数据显示设施

三菱公司开发有多功能简易显示模块（5DM），不通过通信口直接与 PLC 连接即可实现数据显示，其功能较多。OMRON CP1H 型 PLC 面板上也有两个数码管，也可用以显示一些 PLC 的重要信息。

功　能		内　容
时钟功能	显示	显示时钟功能(FX1S、FX1N 内置)的当前时刻
	设定	时间的设定(年,月,日,时,分)
软元件监控功能	位软元件监控	显示 X,Y,M,S 的 ON/OFF
	字软元件监控(16 位)	显示 T,C 的当前值/设定值以及 D 的当前值
	字软元件监控(32 位)	显示 32 位计数器 C 的当前值/设定值以及 D 的当前值
缓冲存储器监控功能		显示特殊模块、特殊程序组的缓冲存储器(仅 FX1N 有效)
出错显示功能		发生可编程控制器出错时,显示出错代码和出错步号
强制设定/重新设定		强制 ON/OFF Y,M,S 的位软元件
T,C 复位功能		清除 T,C 的当前值(当前值:0,接点:OFF)
数据更改功能	当前值更改	更改 T,C,D 的当前值
	设定值更改	更改 T,C 的设定值
保护功能		可以设定操作者功能的所有操作有效、仅监视功能有效、仅时钟时刻显示有效
指定软元件监视功能		可指定 5DM 上显示的软元件种类以及软元件编号
出错显示有效/无效		可以设定操作者功能的出错显示功能的有效或无效
背景光自动熄灭		可以设定背景光的自动熄灭时间(初始值:10 分)
操作键状态		可识别 4 个操作键的 ON/OFF

此外，西门子公司开发的 C7 系列 PLC，就是带有显示与操作界面的 PLC。它把 S7-300PLC 和一个操作员面板（OP）合二为一，既提供了友好的操作界面，又使得整个机器控制系统实现了尺寸最小化，工程造价经济化。

不仅如此，现在还出现有带显示屏的 PLC。相信，随着此类技术在 PLC 上的应用，一个界面友好的，可方便用于数据处理的 PLC 将越来越多地展现在广大用户面前。



图 6-58　ZEN 型 PLC 外观

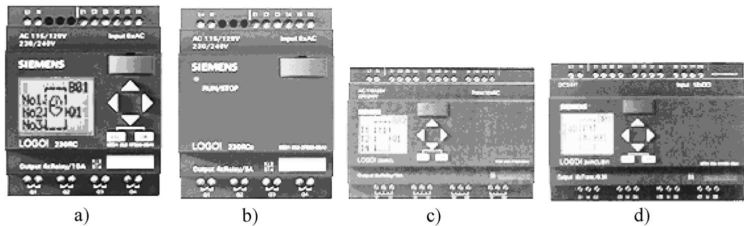


图 6-59　Logo 系列型 PLC 外观

a）LOGO！基本型　b）LOGO！不带显示屏　c）LOGO！加长型　d）LOGO！总线型

最后要提及的是自带有简易编程装置的微型PLC。它点数不多，但输出电流很大，直接可代替继电器控制。这种 PLC 就是典型的带有操作与显示界面的 PLC。图 6-58 所示为 OMRON ZEN 系列 PLC。图 6-59 所示为西门子 Logo 系列 PLC。图6-60所示为三菱公司 Alpha 型 PLC 外形简图。

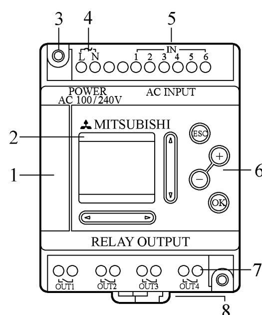


图 6-60 Alpha 型 PLC 外形简图
1—内存盒、编程口端盖 2—液晶显示屏
3—安装孔 4—电源端子 5—输入端子
6—操作键 7—输出端子 8—导轨安装爪

从图知，在它们的面板上，都配置有操作键、液晶显示屏。可通过它用梯形图编程。也可用专用编程软件，在计算机上进行。同时，也可用其显示数据。

有了上述种种数据显示设施，可直接察看 PLC 的内部数据。为使 PLC 实现数据终端的功能提供了很大方便。

6.7 PLC 数据传送

关键词：在线传送、主动传送、被动传送、脱机传送

在数据源和数据宿之间传送数据的过程，也称数据通信。数据源和数据宿的概念是相对的。一般讲，作为数据终端的 PLC 即为这里讲的数据源。而上位机则为数据宿，是 PLC 采集、加工数据的归宿。但有时，PLC 也要执行上位机给予它的指令，完成一些数据显示、打印或报警等工作。

传送数据有两种方法：在线传送及脱机传送。

1. 在线传送

在线传送也就是联网传送。它通过通信口及通信介质，按一定协议传送。有两个方式：被动传送及主动传送。

（1）被动传送。被传送数据或指令由上位机发出，PLC 接到数据或指令后予以响应。什么时候传送，传送什么，由上位机决定。

如本章第一节介绍的分时电量采集，它通过长话线、MODEM，上位链接网，实现远程数据读取（传送），就是典型的被动传送。

被动传送时，虽 PLC 总是工作，但上位计算机较灵活，可不必整天开机。只是需要时开机与 PLC 联网，进行数据传送。

（2）主动传送。被传送数据或指令由 PLC 发出，上位机接到数据或指令后予以响应。什么时候传送，传送什么，由 PLC 决定。

如用 PLC 控制甲设备，而这个甲设备的工作要与乙设备的协调。甲设备与乙设备直接又没能交换数据，但各可与上位机传送数据。这时，PLC 就要用主动通信。对甲设备操作时，要先与上位机传送数据，经确认，才可进行对甲设备的操作。

再如，用 PLC 采集与存储数据，如存储区满，也必须主动把数据传送给上位机传送数据。或向上位计算机提出传送数据要求，待上位计算机确认后再传送数据。这种情况，如不能主动传送，将丢失数据。

主动传送时，PLC 与上位机都要工作，并联网。任何环节出现问题，PLC 的正常控制或数据传递都可能无法实现。

2. 脱机传送

脱机传送用于手工操作。其条件是要有相应的数据载体。

早期 PLC 只有传送程序的载体，如内存卡，可用于 PLC 间脱机进行程序传送。办法是，配备有内存卡的 PLC ，先把程序存入卡中，然后把已存有程序的卡另装到另一 PLC 中，以向另一 PLC 进行程序传送。

还有软磁盘或磁带，也可成为脱机程序传送的载体。它通过简易编程器及附加的软驱或收录机，用人工操作（操作比较麻烦），实现 PLC 间的程序传送。

但用载体传送数据过去是没有的。但现在，由于 PLC 内存及内存卡容量的增大，这两者不仅可存储大量程序，而且也可以文件的形式大量存储数据。这样，可从 PLC 转存到内存卡，并用卡作载体，再转存到别的 PLC 或计算机（新型的内存卡，如 CJ1J 型 PLC 的内存卡，计算机可向其读写数据）。

脱机传送，不仅要有载体，还要人工参与，是费钱费力的方法。所以一般是不用的。但是，脱机传送不用通信设备，系统较简单，数据传送较安全。

PLC 数据传送主要与 PLC 联网、通信有关。故它的有关程序设计将在本书第 7 章进一步讨论。

6.8　数表处理程序设计

关键词：数据加工、求最大、求最小、查找、求和、求平均数

数表是指，在一个存储区中，连续存储的一组数据。数表可大可小。大的可以是某个内存区，如 DM 区。小的可能只有几十个字。在数据采集、存储中，常要用到数表。

为了用好数表，往往要处理数表，如在数表中求最大数、平均数、或对数表排序等等。这些将是本节讨论的内容。

6.8.1　求最大、最小数

1. 用普通指令求

其算法是，指定一个存最大（小）数的字，把它的值与数表中所有的数比较，把比较大者（或小者），留在这个字中。

图 6-61 所示为用普通指令在指定的 DM 区内，求最大数及其 DM 地址的梯形图程序。

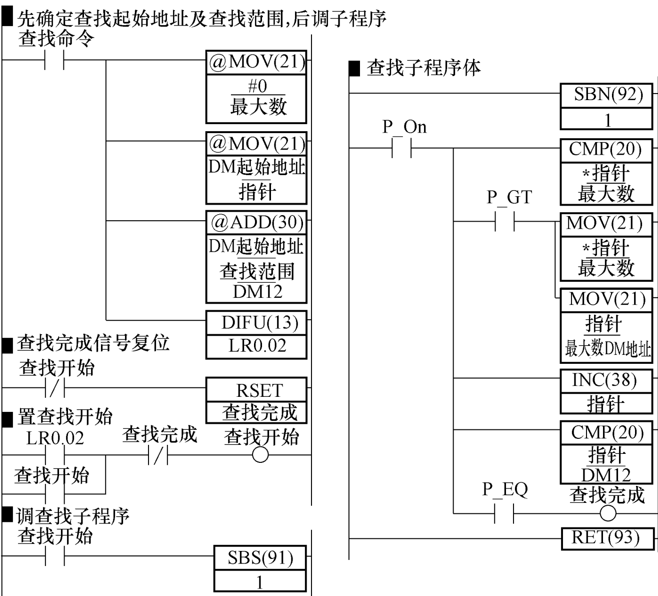


图 6-61　求最大数

该图用的是符号地址。从图梯级 1 知，当“查找命令”ON 时，第 1 条执行的指令 MOV，先使“最大数”置 0（初始化）。接着，把“DM 起始地址”赋值给“指针”，并计算结束地址，结果送 DM12。注意，以上指令都是微分执行的。再就是执行微分指令，使 LR0.02 ON 一个扫描周期。

LR0.02 ON，使图梯级 3 中“查找开始”ON，并自保持，直到“查找完成”的常闭接点 OFF。

在“查找开始”ON 期间，梯级 4 将调子程序 1。每个扫描周期都将调一次。

图中从 SBN 指令开始到 RET 指令之间的程序为子程序。每调一次，总是把“指针”指向的数与“最大数”进行比较，如前者大，则把前者换为“最大数”，并把这个“指针”值送“最大数 DM 地址”。接着，修改指针，并判断是否“指针”已达到最后位置。

到了“指针”值大过 DM12 的值，即最后地址，则“查找完成”ON，其常闭接点 OFF。它将使“查找开始”OFF（如图所示梯级 3），“查找开始”OFF 使“查找完成”复位（如图所示梯级 2）。程序复原。

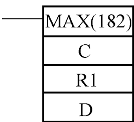
执行这个程序后，“最大数”中存的即为查找出的最大数，“最大数 DM 地址”中存的即为查找出的最大数的 DM 地址。

如要求最小数，则只要把图 6-61 梯形图程序中 P_GT 改为 P_LT 即可，其它的不变。那样，最大数存的就是最小数。最大数地址存的就是最小数地址。

提示：这里求最大、最小数须多个扫描周期才能完成。如用循环指令也可在一个扫描周期内完成。但那样也不好，在查找这个周期，扫描时间可能太长。

2. 用最大、最小值指令求

CJ1 型 PLC 有 MAX（182）指令，可用其查找指定范围内的最大数。指令梯形图格式为



这里　C——控制字，指定了查找范围内的字数；C+1 的第 15 位是 1，数据是有符号二进制数，是 0，数据是无符号二进制数；C+1 的第 14 位是 1，将含有最大值的字的 PLC 内存地址存入 IR00，是 0，不存；

R1——指定了查找范围的第 1 个字，搜索最大值是在从 R1～R1+(C−1) 的范围内的字中进行；

D——用以存放查找范围内的最大值。

例：梯形图程序如图 6-62 所示，其中 C 及 R1 的有关取值如图 6-63 所示。当 00000 ON 时，MAX（182）从 D00200 开始的 10 字范围内搜索最大值。最大值写入 D 即 D00300 中，含有最大值的字的 PLC 内存地址写入 IR00。

图 6-63 所示的是执行过程数据情况。

从图知，在从 D00200 到 D00209 10 个字中，查到最大数是 001B（存于 D00300 中），其在 PC 内存中的地址为 100CA（存于 IR00 中）。

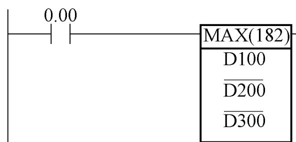


图 6-62　示例程序

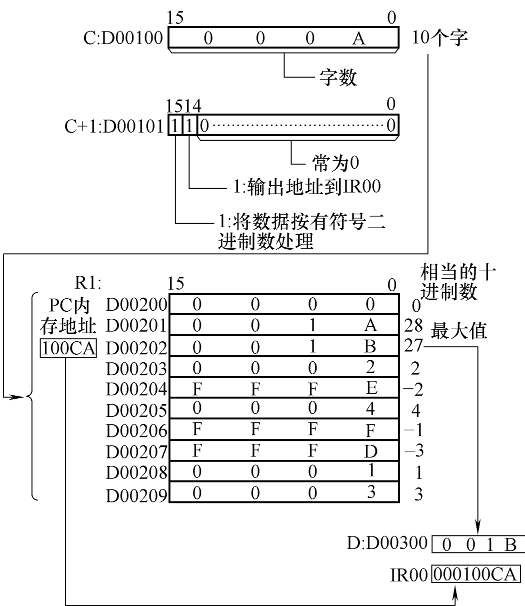


图 6-63　执行过程数据情况

CJ1 型 PLC 有 MIN（183）指令，可用其查找指定范围内的最小数。情况同上。

6.8.2　排序

本排序用的算法为“冒泡法”。图 6-64 所示为这个程序。它用普通指令在指定的 DM 区内，按降幂排序的梯形图程序。

该图用的是符号地址。从图梯级 1 知，当“排序命令”ON 时，第 1 条执行的指令 MOV，使“开始地址”赋值给“指针 0”。接着，又使“开始地址”赋值给“指针 1”。注意，这两个指令都是微分执行的。再就是执行微分指令，使 LR0.05 ON 一个扫描周期。

LB0.05 ON，使图梯级 3 中“排序开始”ON，并自保持，直到“排序完成”的常闭接点 OFF。

在“排序开始”ON 期间，梯级 4 将调子程序 1。每个扫描周期都将调一次。

图中从 SBN 指令开始到 RET 指令之间的程序，为子程序。每调一次，总是把“指针 0”指向的数与“指针 1”指向的数进行比较，如前者小，则把前后两者互换（用 XCHG 指令）。接着，修改“指

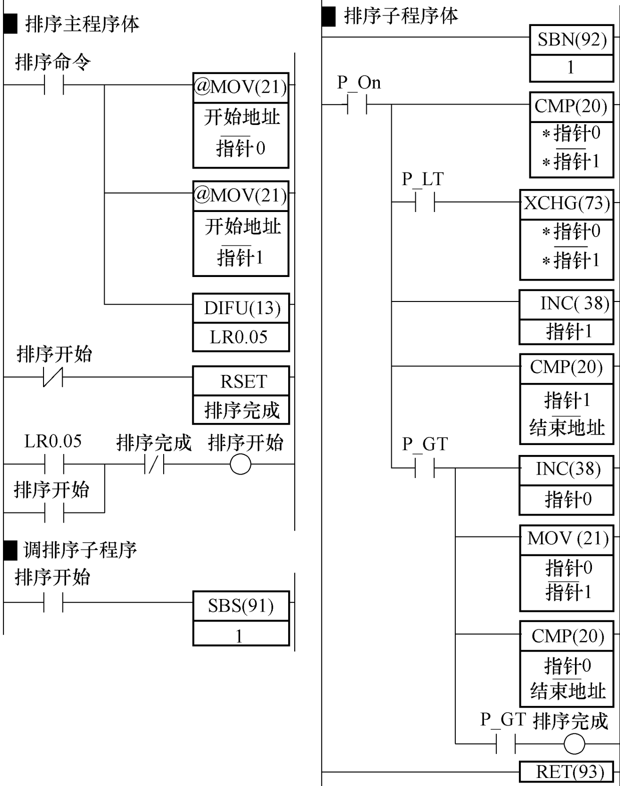


图 6-64　排序程序

针 1”，并判断是否“指针 1”已大于“结束地址”值。

如“指针 1”值大过“结束地址”值，则修改“指针 0”，则把“指针 0“的现值赋值给”“指针 1”。并判断“指针 0”是否已大于“结束地址”值。如不大于，则重复上述循环。如大于，则“排序完成”ON，其常闭接点 OFF，说明排序完成。

“排序完成”常闭接点 OFF，将使“排序开始”OFF（如图所示梯级 3），“排序开始”OFF 使“查找完成”复位（如图所示梯级 2）。程序复原。

执行这个程序后，将使从开始地址到结束地址的 DM 区的数作降幂排序。如要作升幂排序，可把图中 P_LT 改为 P_GT 即可。

提示：这里排序用了两次循环，并须多个扫描周期才能完成。

6.8.3 求总数

1. 用普通指令求

用普通指令求总数，用的基本算法是累加。图 6-65 所示的为在指定的 DM 区内，求总数梯形图程序。

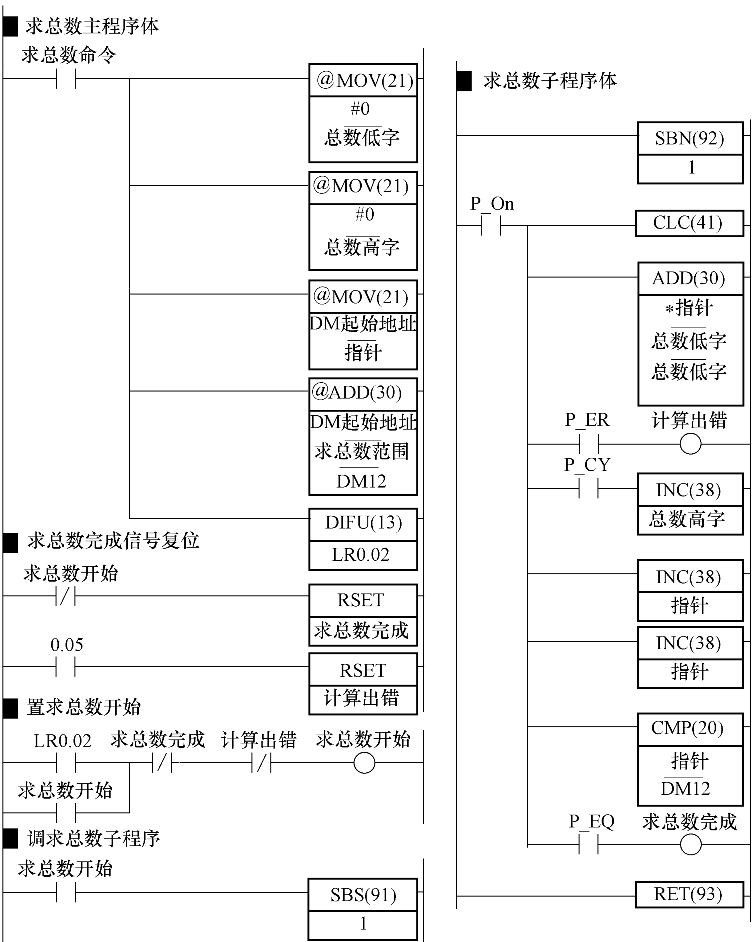


图 6-65 求总数程序

该图用的是符号地址。从图梯级 1 知，当“求总数命令”ON 时，第 1 条执行的指令 MOV，先使“总数低字”置 0。接着，把“总数高字”置 0。再就是把“DM 起始地址”赋值给“指针”，并计算结束地址，结果送 DM12。注意，以上指令都是微分执行的。再就是执行微分指令，使 LR0.02 ON 一个扫描周期。

LR0.02 ON，使图梯级 4 中“求总数开始”ON，并自保持，直到“求总数完成”的常闭接点 OFF，或“计算出错”的常闭接点 OFF。

在“求总数开始”ON 期间，梯级 5 将调子程序 1。每个扫描周期都将调一次。

图中从 SBN 指令开始到 RET 指令之间的程序，为子程序。每调一次，总是先清进位位，再把“指针”指向的数与“总数低字”相加，如有进位，则“总数高字”加 1。接着，修改指针，并判断是否“指针”已达到最后位置。

到了“指针”值大过 DM12 的值，即最后地址，则“求总数完成”ON，其常闭接点 OFF。它将使“求总数开始”OFF（如图所示梯级 4），“求总数开始”OFF 使“求总数完成”复位（如图所示梯级 2）。程序复原。

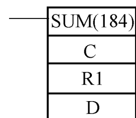
执行这个程序后，求和的值将存于“总数高字”及“总数低字”中。

如求的过程计算有误，如被加数不是 BCD 码，则 P_ER ON，进而使“计算出错”ON。并通过它的常闭接点，使“求总数开始”OFF，计算停止。如再要计算，须先用 0.05 ON，把“计算出错”复位。否则，不能再计算。

提示：这里求总数也需多个扫描周期才能完成。程序中加了出错控制。这是保证数据计算安全的需要。

2. 用 SUM（184）指令求和：

CJ1 型 PLC 有求和，SUM（184），指令，可用其对指定范围内字节或字的数求和，并将结果输出到两个字中。指令梯形图格式为



这里　C——指定求和的单元数（字节或字），C+1的指定单位是字还是字节、数据是二进制数（有符号数或无符号数）还是 BCD 码。如图 6-66 所示说明；

R1——求和范围的首字。

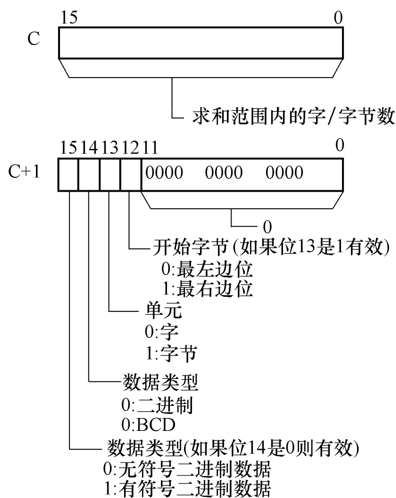


图 6-66　C、C+1 字含义

执行本指令，从 R1 开始到 R1+C 中指定数之间的所有的数相加，并将相加的结果输出到 D+1（存高位）和 D（存低位）中。

例：梯形图程序如图 6-67，其中 C 及 R1 的有关取值如图 6-68。当 00000 ON 时，SUM（184）从 D00100 开始的 10（D00300 内容为 10）字节（D00301 第 13 位为 1）范围内求和。求和的结果写入 D00300 中。

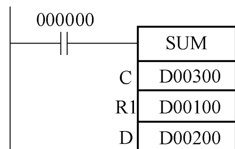


图 6-67　求总数程序

图 6-68 所示是执行过程数据情况。

从图知，在从 D00100 到 D00105 10 个字节求和，结果为 0000037B，存于 D00200 及 D00201 中。

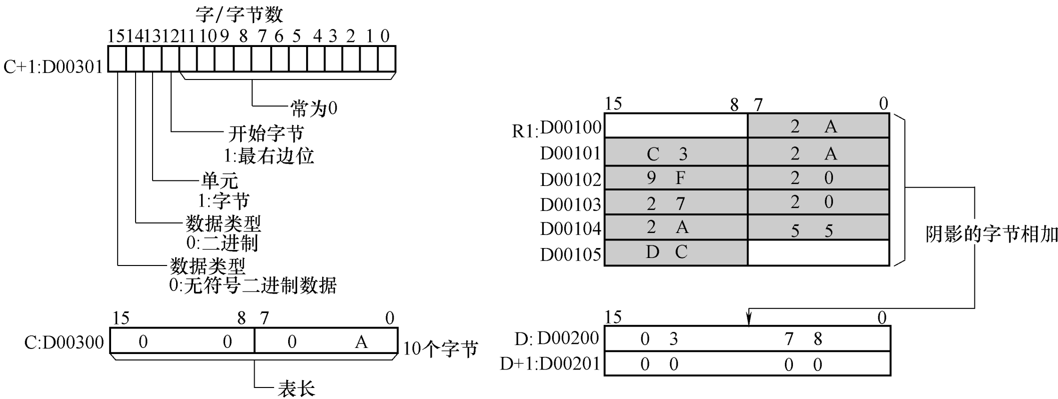


图 6-68 C 及 R1 的有关取值

6.8.4 求平均数

在求出总数后，再除以求总数的范围，就是平均数。故此程序可在图 6-65 的基础上，加入除运算就可以了。图 6-69 所示即为加入的程序。

从图知，只要无计算错误，求总数完成（“求总数开始”ON 后，又 OFF），则进行总数被平均数范围的双字除。结果存于“平均数”中。这里，先把 0 赋值给“求平均数范围高字”，然后再除，以免出错。

提示：OMRON 的双字除指令，被除数只能是地址（间接数）。不能用立即数。

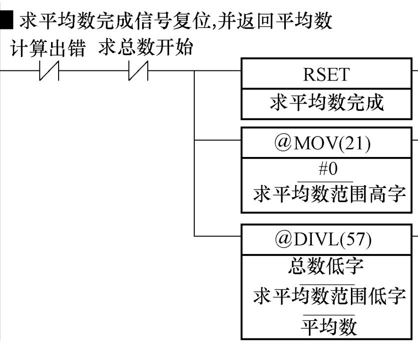


图 6-69 求平均数程序

6.8.5 数据查询

1. 用普通指令求：

其算法是，要查找的数，依此与数表中所有的数比较，发现把比较相等者，就把它的地址记住，留在地址字中。图 6-70 所示为在指定的 DM 区内，查找指定数的 DM 地址的梯形图程序。

该图用的是符号地址。从图梯级 1 知，当“查找命令”ON 时，第 1 条执行的指令 MOV，先使“实际指定数”传“形式指定数”（初始化）。接着，把起始地址赋值给“指针”，并计算结束地址，结果送 DM12。再接着把#FFFF 送“指定数形式地址”。注意，以上指令都是微分执行的。最后就是执行微分指令，使 LR0.02 ON 一个扫描周期。

LR0.02 ON，使图梯级 3 中“查找开始”ON，并自保持，直到“查找完成”的常闭接点 OFF。

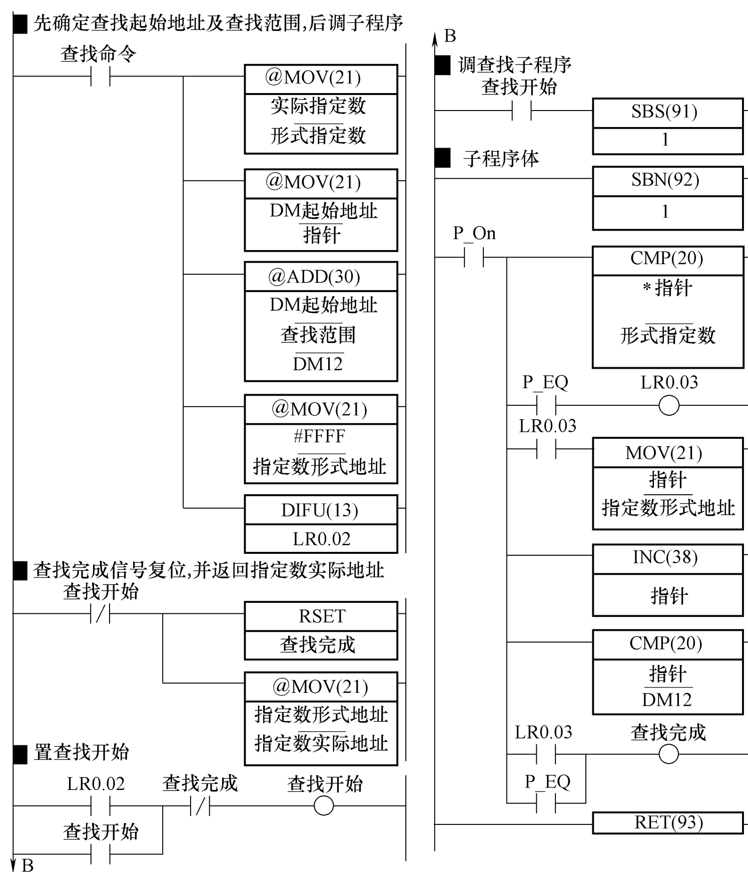


图 6-70　查找指定数

在“查找开始”ON 期间，梯级 4 将调子程序 1。每个扫描周期都将调一次。

图中从 SBN 指令开始到 RET 指令之间的程序为子程序。每调一次，总是把“指针”指向的数与“形式指定数”进行比较，如两者不等，则修改指针，并判断是否，指针已达到最后位置。如相等，则使 LR0.03 ON。而 LR0.03 ON，则把这个“指针”值送“指定数形式地址”，并置“查找完成”ON，表示查找完成。

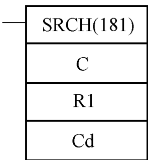
如到了指针值大过 DM12 的值，即最后地址，还没有相等的，也将使“查找开始”OFF（如图所示梯级 3），“查找开始”OFF 使“查找完成”复位（如图所示梯级 2），并把“指定数形式地址”赋值给“指定数实际地址”，程序复原。

执行这个程序后，如找到该数，其地址存于“指定数实际地址”中。如没找到，则“指定数实际地址”值将是 FFFF。

提示：这里查找也须多个扫描周期才能完成。用“指定数实际地址”是否为 FFFF，可知是否找到该数。

2. 用数据搜索指令求：

数据搜索，SRCH（181）指令，用以在指定范围的字中查找某字。梯形图格式为



这里 C——控制字，指定了查找范围内的字数；C+1 的第 15 位为 1，向 DR00 送匹配的数，为 0 ，不送；

R1——指定了查找范围的第 1 个字，搜索是在从 R1～R1+(C－1）的范围内的字中进行；

Cd——比较数据，即要查找的数。

执行本指令，将在 R1 到 R1+(C－1）的内存范围内，搜索含有 Cd 的字。如找到一个匹配字，则将该字的 PLC 内存地址写入 IR00，并将等于标志置为 ON。如果有 2 个或更多的匹配字，只有第一个含有比较数据的字被写入 IR00。

例：程序如图 6-71 所示，这里控制字的内容为 8000000A，即搜索范围为 10 个字，并要把匹配的数送 DR00。

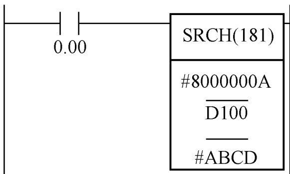


图 6-71 查找数据

从图知，当 0.00 ON 时，即执行 SRCH 指令。在从 D100 开始的 10 字范围中，搜索与 ABCD 相同的字。如仅 D103 的内容为 ABCD，则 IR00 内容为 00010067（D103 的内存地址），而 DR00 的内容为 1（仅一个数匹配）。

可见，有了这么强的指令，做数表处理是很简单的。

结语

PLC 用作数据终端可在实施控制的同时，实现现场数据采集、存储，为实现信息化基础上的自动化创造前提条件。为 PLC 要好的地实现数据终端功能，解决数据输入、显示是重要的。尽管由于人机界面等开发已有不少进步，但似乎还必须探讨既要界面友好，又要系统简单的解决方案。

在本章介绍的算法实例化中，用了多扫描周期子程序执行的方法，还用了一些较复杂的间接寻址及运算指令等。所以，在本章的学习中，还可加深对第一章的内容的理解。

请想想

1. PLC 数据处理的含义？
2. 用 PLC 作数据终端的好处？
3. PLC 用作数据终端的条件？
4. PLC 采集数据的类型及可行方法？
5. PLC 录入数据的可能方法？
6. PLC 存储数据的类型及方法？

7. 显示 PLC 内部数据的重要性及方案？
8. 多扫描周期子程序执行的意义？

请试试

1. 编写数据压缩程序
2. 用其它排序算法编写排序程序
3. 编写 5 个数的加权平均程序
4. 其它实际控制程序设计

第 7 章　PLC 通信编程

本章将讨论 PLC 与 PLC、PLC 与计算机、PLC 与人机界面以及 PLC 与其它智能装置间的通信编程。

7.1　概述

关键词：PLC 链接、PLC 网络、信息网、控制网、设备网、监控与数据采集（SCADA）、人机界面（HMI）

7.1.1　PLC 通信的目的

PLC 可与计算机、人机界面、PLC、智能装置通信，各有其目的。

1. PLC 与计算机、人机界面通信的目的

（1）实现计算机控制系统的监控与数据采集（Supervisory Control And Data Acquisition SCADA)，是实现在信息化基础上自动化的一个重要工作。具体是：

1）读取 PLC 工作状态及 PLC 所控制的 I/O 点及内部数据的状态，并显示在计算机的屏幕上，以便于人们了解 PLC 及其控制设备的工作状态。

2）通过向 PLC 写数据，改变 PLC 所控制的设备工作状况，或改变 PLC 的工作模式，实现人们对控制的必要干预。

3）读取由 PLC 所采集的数据，并进行处理、存储、显示及打印，便于人们使用现场数据。

（2）实现人机界面的监控及管理。人机界面（Human Machine Interface，HMI）可显示数据，又可输入数据，具有信息处理、采集及显示功能，近年来已用得越来越多。它与 PLC 联网通信，可从 PLC 读取数据，并予以显示。也可向它写数据，再传输给 PLC，改变 PLC 的状态或数据，实现对 PLC 或系统的控制。

虽然 PLC 的功能不如计算机，但它的体积小、工作可靠、可安装在工业现场，所以用起来是很方便的，在一定程度上，也可起到计算机的 SCADA 作用。

（3）实现 PLC 编程及程序调试。PLC 编程是较麻烦的。若用手持编程器，通过助记符编程则更麻烦。但若用计算机与 PLC 联网，再使用相应编程软件，则可使用梯形图或流程图语言编程，以至于还可用其它高级语言编程，则较方便。

而且用计算机编程，还可对所编的程序进行语法检查，便于发现与查找程序中的语法错误同时还可进行程序仿真，可对输入点的状态进行强制置位或复位，可模拟现场情况，进而可发现与解决程序中语义方面的问题。

此外，计算机编程还可存储、打印 PLC 程序，或把程序写入 ROM 中等，便于 PLC 程序的移植及重用。

所以，使用与计算机联网，进行 PLC 编程已是一个趋势。有的厂家的高级 PLC 编程器，

实质上就是笔记本式个人计算机。

（4）实现远程诊断与维护。有的 PLC 配备有远程诊断与维护系统。可通过联网，甚至互联网，利用厂商提供的通信模块与专用软件，实施远程数据采集与故障诊断，并实施相应维护，如下载新版本的 CPU 操作系统、硬件驱动程序等。

2. PLC 与 PLC 通信的目的

（1）扩大控制地域及增大控制规模。PLC 多安装于工业现场，用于当地控制，但如果进行联网通信，则可实现远程控制，距离近的可以为几十、几百米，远的可达几千米，或更远，可大大扩大 PLC 的控制地域。

联网后还可增加 PLC 控制的 I/O 点数。这里，尽管每台 PLC 控制的 I/O 点数不变（有的 PLC，加远程单元后，也可增加 I/O 点数），但由于联网后，为多台 PLC 参与控制，其控制总点数为参与联网的 PLC 控制点数之和，显然，其控制的规模要比单个 PLC 的规模大。

不少事实说明，两个或若干个中、小型 PLC 联网，也可达到一个大型机的控制规模，而费用则比大型机要低得多。因而，用中、小型 PLC 联网，去替代大型 PLC，已成为一个趋势。

（2）实现系统的综合及协调控制。用 PLC 实现对单个设备的控制是很方便的。但如果有若干个设备要协调工作，用 PLC 控制，较好的办法是联网通信。单个设备各由各的 PLC 控制，而设备间的工作协调，则依靠联网通信后 PLC 间的数据交换解决。

这样联网通信，可使 PLC 控制，从设备一级提升到生产线一级，便于实现工厂的综合自动化。

（3）提高 PLC 工作的可靠性。联网通信后，各 PLC 还是独立工作的。只要协调好，个别站点出现故障，并不影响其它站点工作，更不至于全局瘫痪，故可降低系统的故障风险。

（4）联网通信更便于实现冗余配置。工作 CPU 与热备用 CPU 使用联网通信处理转换，用起来非常方便，系统也非常可靠。是当今冗余配置的一个趋势。

3. PLC 与智能设备通信的目的

（1）简化系统布线。工业现场的普通开关量及模拟量输入、输出信号，都是通过信号线与 PLC 的 I/O 点相连，连线很多，布线也很复杂，特别是一些远离 CPU 的输入、输出点多时，则更为复杂。而如果用联网通信传送输入、输出信号，则简单得多，通信线仅用两根或几根，就可传送很多输入、输出信号。尽管这样处理有一定延时，但由于网络传送及信号处理速度的不断提高，这点延时，对一般的控制过程是不会有什么影响的。

（2）便于系统维修。输入、输出信号用联网通信处理，布线简单了。这样既可节省硬件开支，还便于系统维修。试想，一大堆的输入、输出接线，真有个别出现问题，在现场查找是很不易的。

（3）实施对智能设备的管理。智能设备都有自身的 CPU、内存及通信接口，自身可采集、处理或使用数据。但是，如果没有联网通信，它们还只是“信息孤岛”，无法与其它控制协调，或实现信息共享。而如果与 PLC 联网通信，则可使这些装置将不再是“信息孤岛”了。

特别要强调的是，信息化已是当今信息社会的潮流。已给世界带来了巨大的经济效益与社会效益。而企业的信息化，推行企业资源计划（ERP）、信息执行系统（MES），以至于产品生命周期计划（PLM），更是给企业带来了不可估量的效益。

而企业信息化的基础是获得准确的生产一线的数据。PLC 联网通信后，这些数据可自动采集，避免了人为干扰，因而是客观的、准确的，而且是完整的、及时的。这就为建立智能工厂、透明工厂、全集成系统或 e 自动化等取得成功提供了基础。

总之，PLC 联网通信从根本上来说主要是用好信息，目的是协调控制，进而实现系统控制的自动化、远程化、信息化、智能化。

7.1.2　PLC 通信平台

PLC 与计算机、人机界面 PLC 或智能设备通信最简便的是利用标准串行接口。而理想平台是 PLC 网络。

1. 标准串行接口

用以将并行数据转换为串行数据，以实现串行通信。有 RS-232C、RS-422A、RS-485 及 USB 接口。

（1）RS-232 接口。RS-232 接口是指合乎 RS-232 标准的串行通信接口。RS-232 标准（协议）的全称是 EIA-RS-232 标准，是 1960 年，由美国电子工业协会（Electronic Industry Association，EIA）制定的。RS（Recommended Standard）代表推荐标准，232 是标准的标识号。以后又陆续发布了修订版本 RS-232A、RS-232B 和 RS-232C。目前广泛应用的是 RS-232C，为最新一次修改。

RS-232 标准定义的是，数据终端设备（Data Terminal Equipment，DTE）和数据通信设备（Data Communication Equipment，DCE）之间串行二进制数据交换接口的技术标准。DTE 可以是计算机，也可是 PLC 或智能设备，指需要数据通信的站点。DCE 是用以实现站点间通信的设备，如调制解调器（Modem）等。它的典型应用如图 7-1 所示。

从图 7-1 可知，有了 DCE 这个通信设备，两个 DTE（站点）才可建立物理连接，进而也才可进行数据通信。

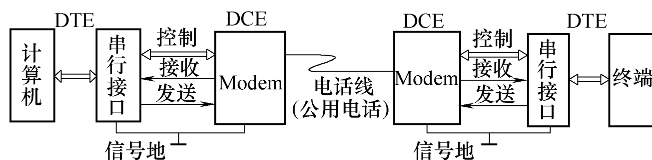


图 7-1　RS-232C 串行接口典型应用

DCE 也还可以是无线调制解调器。在各无线调制解调器间，可通过无线电波或红外线传送信号，也可达到数据通信的目的。

1）RS-232C 接口特性。RS-232C 标准对串行接口规定有，电气特性、机械特性、功能特性及规程特性，并对其多有所推荐，没有推荐的也有俗成约定。只是这个标准毕竟是美国的，而且年代又较久远，所以，在使用时，还要注意到对其理解与实施时的差异。

（a）电气特性。电气特性规定信号电平，电路连接方式，串行数据的传输格式、速率、距离，以及与互联电缆相关的规则等。

RS-232C 标准规定有 2 条数据、多条控制信号线及一条共用信号地线，适用于 1 对 1 全双工通信。

RS-232C 标准规定，信号为负逻辑。在数据线及控制线上，逻辑 1 为 −3V ~ −15V，逻辑 0 为 +3 ~ +15V。输入的逻辑电平是 +3V ~ +15V 和 −3V ~ −15V。输出为 +5V 到 +15V和 −5V ~ −15V。输入与输出不同，主要是考虑到抗干扰裕度，即 DTE 与 DCE 间允许有 2V 压降。

RS-232C 标准规定，数据按字符（节）为单位传送。每个字符（节）用起始位及使用相同的发送接收速率，保证收发同步。

（b）功能特性。功能特性规定了接口引脚功能和接线。当初，它规定了 21 条信号线，但在这 21 条信号线中，常用的仅有 9 条，也因此 9 引脚连接器就很常用了。这 9 针连接器引脚的信号线定义见表 7-1。

表 7-1　9 针 RS-232C 接口引脚定义

引脚号(9 针)	信　号	方　向	功　能
1	DCD	IN	数据载波检测
2	RxD	IN	接收数据
3	TxD	OUT	发送数据
4	DTR	OUT	数据终端装置(DTE)准备就绪
5	GND		信号公共参考地
6	DSR	IN	数据通信装置(DCE)准备就绪
7	RTS	OUT	请求传送
8	CTS	IN	清除传送
9	CI(RI)	IN	振铃指示

（c）机械特性。机械特性规定物理连接的插头和插座的几何尺寸、插针或插孔芯数及排列方式、锁定装置形式等。但 RS-232C 标准并未对机械接口作出严格规定。目前常用的 9 引脚数据总线连接器（插头、插座）简图如图 7-2 所示。

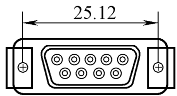


图 7-2　9 引脚接口简图

插针或插孔连接器的几何尺寸是相配合的，互成镜像对称。连接器的 DTE 端为针，DCE 端为孔，与它相接的为针。一般来说，作为 DTE 的计算机的连接器为针，而 PLC、智能设备同样也是 DTE，但使用孔。

（d）规程特性。规程特性规定串行接口通信的实现过程或步骤。与是否通过 DCE 有关。一般总是要先拨号，要求建立连接；经对方回应确认，实际建立连接；然后才可通信。通信后还要挂机，断开连接。图 7-3 所示为计算机串行接口与 PLC 串行接口，通过 DCE（调制解调器，Modem）连接示意。

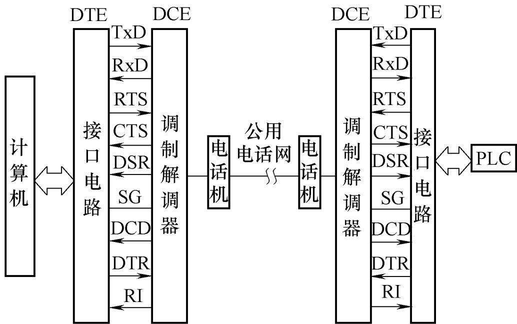


图 7-3　计算机与 PLC 通过 DCE 连接示意

对这样的连接，如果计算机发起通信，则要把 PLC 方的 Modem 置为准备接收呼叫状态，办法是，把 PLC 方的 Modem 接于计算机串行接口，调 Windows 附件中的超级终端，在其窗口上键入以下三条命令：

atso =1（设置准备接收状态）

at&wo（存于寄存器）

at&yo（上电时调寄存器）

这样，Modem 接 PLC，上电后，它的 AA 指示灯亮。这表明 PLC 方 Modem 已作好接收呼叫的准备。这时，一旦计算机呼叫它，经一声振铃响，即响应，并抢占电话线。

计算机方要发起通信，则要在用户程序中，执行与 PLC 方 Modem 通信的呼叫代码。也

可通过超级终端键入呼叫代码。呼叫的代码是：

atdt + PLC 方电话号码 + 回车

一旦呼叫成功，将返回相应信号，双方 Modem 的 CD 指示灯均亮。这时，计算机与 PLC 完成了通信连接。其间虽经过 Modem、电话局，但将如同直接连线，即可通信。

当通信距离较近时，可不需要 DCE，通信双方可以直接连接。有两种连接方式：一为有握手信号连接；另一为无握手信号连接。后者更常用。

有握手信号连接除了双方数据交叉相接、信号地线直接相连外，其它的信息线也要相应地进行连接。较完整的连接如图 7-4 所示。

这种连接称为交叉环回接口连接，也称假 Modem 连接。双方需用 7 条线，数据交换前要进行“握手”。“握手”成功才可交换数据。双方“握手”信号过程如下：

a）当甲方的 DTE 准备好，发出数据终端准备好（DTR）信号，该信号直接连至乙方的 RI（振铃信号）和 DSR（数传机准备好），即只要甲方准备好，乙方立即产生呼叫（RI）有效，并同时数传机准备好（DSR）。尽管此时乙方并不存在 DCE。

b）甲方的 RTS 和 CTS 相连，并与乙方的 DCD 互连。即一旦甲方请求发送（RTS），便立即得到允许（CTS），同时，使乙方的 DCD 有效，即检测到载波信号。

c）甲方的 TxD 与乙方的 RxD 相连，一发一收。

图 7-5 所示为零 Modem 方式的最简单连接（即三线连接），图中的 2 号线与 3 号线是交叉连接。这样，把通信双方都当作数据终端设备看待，双方可发，也可收。通信双方的任何一方，只要请求发送（RTS）有效和数据终端准备好（DTR）有效，即能开始发送和接收。

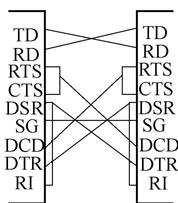


图 7-4　RS-232C 接口有握手信号连接

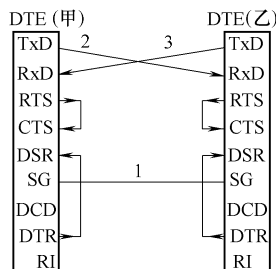


图 7-5　RS-232C 口无握手信号连接

2）RS-232C 的不足之处。RS-232C 接口信号电平值较高，易损坏接口电路的芯片；只能进行一对一的连接，不能实现与多个站点通信；使用一条共用信号地线，共地传输，易产生共模干扰，所以抗噪声干扰性弱。传输距离（电缆长度）是受限制的。当传输速率为 19200bit/s，误码率小于 4% 时，要求导线的电容值应小于 2500pF。对于普通导线，其电容值约为 170pF/m。所以，其最大传输距离可按下式计算：

$$允许距离\ L = 2500\text{pF}/(170\text{pF/m}) = 15\text{m}$$

当使用 9600bit/s，普通双绞屏蔽线时，距离可达 200ft（60.96m）。当使用 1200bit/s，普通双绞屏蔽线时，距离可达 3000ft（914.4m）。

（2）RS-422A 接口。为了克服 RS-232 标准的不足，于 1977 年底，EIA 颁布了新的接口标准。几经改进，后出现了 RS-422A 标准。规定不使用控制信号；规定的信号电平降低为 ±6V（±2V 为过渡区域，仍为负逻辑）；规定发送器、接收器分别采用平衡发送器和差动

接收器。这相当于差动直流放大器可消除零点漂移一样，消除共模干扰，所以抗串扰能力大大增强。

图 7-6 所示为 RS-422A 四线接口电气原理图。除 4 条信号线外，还有一条信号地线。数据线发送端（SDA、SDB）与对方的接收端（RDA、RDB）相连，各有其通道，无须握手直接可以进行双向数据交换。

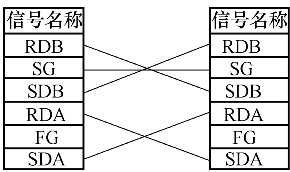


图 7-6　RS-422A 四线接口电气原理图

RS-422A 的接收器输入阻抗较高（4kΩ），发送器的驱动能力也比 RS-232C 大，所以它允许一个发送器可连接多到 10 个节点。但其中只有一个主站点（Master），其余为从站点（Slave），而且从站点之间不能通信。

RS-422A 平衡双绞线的长度与传输速率成反比。最大传输距离为 4000ft（约 1219m），最大传输速率为 10Mbit/s。一般 100m 长的双绞线上所能获得的最大传输速率仅为 1Mbit/s。

RS-422A 连线时，在传输电缆的两个最远端，要接终端电阻，要求其阻值约等于传输电缆的特性阻抗。但距离较短，如 300m 以下时，一般可不接。

（3）RS-485 接口。RS-485 是从 RS-422A 基础上发展而来的。它的许多电气性能与 RS-422A 相仿。如都采用平衡传输方式，在传输线终端接终端电阻的要求也与 RS-422A 相同。但 RS-485 连线即可用 4 线制，也可用 2 线制。

采用 4 线连接也只能有一个主（Master）站点，其余为从站点。从站点之间也不能通信，但从站数可增加到 32 个。

RS-485 用的更多的是 2 线制，采用半双工方式，靠使能控制达到双向通信的目的。其接线如图 7-7 所示。在一个总线上，它可允许连接多达 128 个 RS-485 接口（站点），而且站点间可实现相互通信。

从图 7-7 可知，在 RS-485 中有一个“使能”端（该图未全画出），用于控制发送驱动器和传输线的切断与连接。哪个站点要发送数据，则用“使能”端激活，而其它站点则只能接收数据。

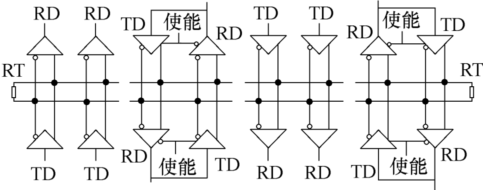


图 7-7　RS-485 接口连线

RS-232C、RS-422A 及 RS-485 有关电气特性参见表 7-2。

表 7-2　RS-232C、RS-422A 及 RS-485 有关电气特性

规　　定	RS-232C	RS-422A	RS-485
工作方式	单端	差分	差分
节点数	1 收 1 发	1 发 10 收	1 发 32 收
最大传输电缆长度/ft	50	400	400
最大传输速率/(Mbit/s)	0.02	10	10
最大驱动输出电压/V	±25	−0.25 ~ +6	−7 ~ +12
驱动器输出信号电平（负载最小值）/V	±5 ~ ±15	±2.0	±1.5
驱动器输出信号电平（空载最大值）/V	±25	±6	±6

（续）

规　　定	RS-232C	RS-422A	RS-485
驱动器负载阻抗/Ω	3000	100	54
摆率(最大值)/(V/μs)	30	N/A	N/A
接收器输入电压范围/V	-15 ~ +15	-10 ~ +10	-7 ~ +12
接收器输入门限电压/V	±3	±0.2	±0.2
接收器输入电阻/kΩ	3 ~ 7	4(最小)	≥12
驱动器共模电压/V		-3 ~ +3	-1 ~ +3
接收器共模电压/V		-7 ~ +7	-7 ~ +12

提示：由于 RS-485接口的优越特性，有的 PLC 专用网络的物理层，如 Profibus、CC-Link 网，也是用 RS-485串行接口标准。

（4）USB 接口。除了串行接口，还有 USB（Universal Serial Bus，通用串行总线）接口及 IEEE 1394 也是使用串行传输方式传送数据。

通用串行总线（USB）是 1995 年康柏、微软、IBM、DEC 等公司为解决传统总线不足而推广的一种新型的接口标准。目前有两个版本，USB1.1 的最高数据传输速率为 12Mbit/s，USB2.0 则提高到 480Mbit/s。所有支持 USB 1.1 的设备都可以直接在 USB 2.0 接口上使用。USB 连接线为设备还可提供最高 5V、500mA 的电力。

USB 接口可热插拔，其连接的外设可即插即用。此外，USB 接口还具有体积小、安装方便、高带宽、价位低、易于扩展、可连接多台外设（最多可连接 127 台），以及可在线供电、不占系统资源、可错误检测与复原、节省能源等优点，所以 USB 接口已经成为计算机的标准接口之一。

正是 USB 接口有这么多的优点，最新出厂的 PLC，如 OMRON 公司的 CP1H 型 PLC，也开始配置有这个接口，可使用它通过厂商所提供的编程软件，与计算机直接通信。

USB 接口的数据传输距离不大，超过 5m 便不能通信。

除了 USB 接口，还出现 IEEE 1394 接口。它的前身叫 Firewire（火线），是 1986 年由苹果电脑公司针对高速数据传输所开发的一种数据传输接口，并于 1995 年获得美国电气与电子工程师学会认可，成为正式标准。现在看到的 IEEE 1394、Firewire 和 i.LINK 其实指的都是这个标准。

IEEE 1394 也是一种高效的串行接口标准，功能强大而且性能稳定，支持热插拔和即插即用。IEEE 1394 可以在一个端口上连接多达 63 个设备，设备间采用树形或菊花链拓扑结构。

IEEE 1394 是横跨计算机及家电产品平台的一种通用接口，适用于大多数需要高速数据传输的产品，如高速外置式硬盘、CD-ROM、DVD-ROM、扫描仪、打印机、数码相机、摄影机等。

相比于 USB1.1，IEEE 1394 接口在速度上占据了很大的优势。而当 USB2.0 推出后，IEEE 1394 接口在速度上的优势就不再那么明显了。同时现在绝大多数主流的计算机并没有配置 IEEE 1394 接口，但可以通过插接 IEEE 1394 扩展卡的方式获得此接口。本书出版之前，还没有配置这样接口机型的 PLC。

IEEE 1394 接口工作速度虽快，但传输的距离更短，最长仅 4m 左右。

（5）通信接口转换。从以上介绍可知，标准通信接口种类很多。不同类型的接口，其电气、功能机械及工作特性也不相同。所以，要实现通信，接口类型不同，通信是无法实现的。

但是为了在不同类型接口之间也能通信，可对其中的一方的接口转换为与另一方的相同。转换后再连接，通信就有可能了。如当今的“笔记本式个人计算机”只配置 USB 接口，把它转换为 RS-232C 接口，就可与配备有 RS-232 接口的 PLC 通信。

再就是，由于计算机要接入通信设备不断增多，有时可能出现通信接口不够用的情况。这时，计算机还可配置如以太网卡那样的接口卡。配置后，其上就有相应标准接口。一个板卡提供几个、十几个接口的都有。

可知，有了转换器、板卡，计算机的标准接口可以转换，也可以增多。而且现在开发、生产这样转换器、板卡的厂商很多。市场上这类产品也很多，价格也较为低廉，这就为计算机通过标准接口实现与多种设备的连接、通信提供了很大方便。

除了转换器，还有标准串行接口的“增强器”。用它可提高标准接口的通信能力，如增大通信距离，增强抗干扰或抗雷击能力。这类“增强器”名称不一，或称适配器，或称收发器，或称串行接口泵。如 RS-232C 接口，原来最大传输距离只有 15m、波特率最高只有 19200bit/s，而经使用串行接口泵“增强”，其传输最大距离可达 2km，波特率也可达 57600bit/s。这些增强技术的奥秘是，将 RS-232C 接口数据传送方式转换为“浮地隔离”双端平衡传输。同时，采用了光隔离，使得串行接口与设备之间没有电连接，只有光传送，大大地提高了系统的抗干扰能力，内置的抗雷击电路也可确保系统的安全。

连接串行接口的媒体本来是使用双绞线，也可用适配器“增强”，变为使用光纤。很多 PLC 厂家，如 OMRON 就生产有这类适配器。使用这类适配器后，设备间用光纤连网通信，不仅可增大网络站点间的距离、避免雷击、提高保密性，同时也可提高传输速率。有的产品还介绍说，经使用“光纤”产品“增强”，传输距离可扩大到 20km。

再就是还有种种传输用的“中继器”。它可把收到的，已减弱的信号放大，放大后再向前传送。所以，它也是接口通信能力“增强”用的。特别在使用 RS-485 时，在站点间增加多个中继器，可大大增加数据的传输距离，同时，还能增多网络站点的配置。

（6）串行接口设定。使用串行接口还要做好设定，也就是要确定它的波特率、帧格式及校验处理。波特率控制数据发送与接收速率，只有通信双方传输速率相等，所发送的数据才能被正确接收。帧格式是指起始位、数据位及停止位的分配。常用的是 1 位起始位，7 或 8 位数据位，1 到 2 位停止位。校验有奇或偶校验，也可设为不用校验。帧格式及校验处理通信双方也要设定一致。注意，如果传输的为十六进制数，而不是 ASCII，则必须用 8 位数据位。

此外，还有数据流的控制。因为接收方在收到数据后，总是要进行处理的，如果接收方处理速度较慢，也有可能未把所接收的数据处理完毕，接收缓冲区还未清空，发送方又把数据发来，这样就要造成数据丢失，这当然是不允许的。流控制就是接收方通过控制信号通知发送方是否发送或停止发送数据。

流控制有两种办法：硬件流控制及软件流控制。

硬件流控制：硬件流控制常用的有 RTS/CTS（请求发送/允许）流控制和 DTR/DSR（数据终端准备好/数据设置就绪）流控制。为此，必须将相应的电缆线连上。在编程时，

要根据接收端缓冲区大小设置一个高位标志（可为缓冲区大小的75%）和一个低位标志(可为缓冲区大小的25%)，当缓冲区内数据量达到高位时，在接收端将 CTS 置逻辑 0，当发送端的程序检测到 CTS 为逻辑 0 后，就停止发送数据。当到接收端缓冲区的数据量低于低位时，再将 CTS 置逻辑 1，这时发送端的程序检测到 CTS 为逻辑 1 后，又继续发送数据。

软件流控制：由于有的串行接口不使用控制信号线，流控制可用软件流实现。常用方法是：当接收端的输入缓冲区内数据量超过设定的高位时，就向数据发送端回应 XOFF 字符(十进制 ASCII 值 19)，而发送端接收到“XOFF”字符后，就停止发送数据。当接收端的输入缓冲区内数据量低于设定的低位时，就向数据发送端回应“XON”字符（十进制的 ASCII 值 17)，而发送端接收到“XON”字符后，继续发送数据。

应该注意，若传输的是二进制数据，标志字符也有可能在数据流中出现而引起误操作，这是软件流控制的缺陷，而硬件流控制不会有这个问题。

串行接口的设定多数是用软件在串行接口初始化时实现。有的设备也可使用硬件开关选定。

2. PLC 网络

（1）PLC 网络概念。PLC 网络是指分布在不同地理位置上各自独立工作的 PLC、计算机或智能设备，通过通信组件与通信介质，在物理上相互连接在一起，并在网络软件及通信程序管理与协调下实现数据通信的系统。

在 PLC 网络中各个独立工作的 PLC、计算机、人机界面或智能设备称为网络站点，也称节点或结点。

PLC 网络站点的多少，取决于网络的规模，多的有几十，甚至更多；少的只有十几个、几个，以至于仅两个。如仅拥有两个站点的网络有时也称链接。

PLC 网络还可互联。指的是，通过网关、网桥，即同时连接在不同网络上的站点或网络组件，使不同网络互联，从而可使在不同网络的站点之间，也可实现数据通信。网络互连后，仍然还统称为网络。

PLC 网络要有如下三个“应有”：

1）应有物理连接，以在站点间建立通信通道。这是显然的，没有物理连接，没有通信通道，站点间不能通信，当然不是什么 PLC 网络。为此要做到：

（a）要选好通信媒体或介质。通信介质可以是有线的，如电缆、光缆；也可以是无线的，如无线电波、光波、红外线。类型很多，到底应选用什么介质，则与网络的站点最大距离、网络站点数目以及工作要求及环境有关。

（b）要选用好网络组件，主要是配置与使用好网络模块或接口，以便用它通过通信介质实现站点间的物理连接。

对计算机和人机界面，如用串行接口联网，可使用串行接口；如果接入其它类型的 PLC 网络，那就要插入相应的网卡。

对 PLC，如用串行接口联网，可使用 CPU 单元上的外设接口或串行接口，也可用串行接口模块；如果连接其它类型的网络，那就要配置网络模块。

对智能设备，如用串行接口联网，可使用串行接口；如果要连接其它类型的网络，那就要配置相应的专用网络接口。

当网络站点之间的距离很大时，为确保数据正确传输，需增设相关通信组件，以进行信

号放大或数据中转。而为了网络互联，则还要有相应的网络组件。

2）应有数据通信，以进行站点间信息交流。

对于 PLC 网络，信息主要有两个方面：一是状态信息，包含自身工作及控制对象工作的状况；二是命令信息，包含通信命令发送、应答及工作命令下达。

状态信息交流可使各个 PLC 站点的控制得以协调，进而提高控制能力。同时，状态信息，特别是所收集的对象状态信息，还可上传，为企业的信息管理提供原始数据。

命令信息的传送及应答，是各站点相互操作的需要，也是 PLC 网络所特有的。如用 PLC 控制变频器的输出频率就可用传送命令信息实现。再如计算机控制 PLC 工作，或 PLC 主站控制其从站工作，用的也是命令信息。

进行数据通信、信息交流要做到以下两点：

（a）要有网络管理软件，以进行网络配置与管理。对 PLC 网络，PLC 厂商的编程软件多可进行这个配置与管理，如站点编址、通信参数设定、使用数据区指定等。此工作也称为网络组态或设定，是使用网络必须做好的工作。此外，有的厂商、有的网络还提供专门网络管理软件，也可用以对网络的组态及管理。

（b）要有通信程序，以进行通信数据的准备、传送、接收、处理及使用，进而达到交流信息的目的。对 PLC 网络，这个程序必须由用户编写，要依据控制进程以及交流信息的需要编写。

3）应有 PLC 站点，而且各站点都能独立工作，PLC 应该是核心。网络的数据通信主要是在 PLC 与计算机、PLC 与 PLC 以及 PLC 与智能设备间进行。而在这三种通信中，惟一不能缺少的就是 PLC。

各站点都能独立工作也是必要的。这样，即使网络数据通信瘫痪，但由于各站点都能独立工作，都可实施自身的控制，进而还可保证系统工作的安全。

（2）OMRON PLC 网络，大体上分有 3 级，即企业级（主要用于 PLC 与计算机联网，一般为以太网）、车间级（主要用于 PLC 与 PLC 联网）、现场级（主要用于 PLC 与现场设备联网）。

只是，这里的“级”与 OSI（Open System Interconnection，开放系统互连的体系结构）中网络参考模型的“层”是不同的，后者是网络结构的层。它建议用 7 层，即物理层（规定使用互连电路、电气特性及连接器的配置等）、数据链路层（规定信息基本单元的封装格式）、网络层（确定源及目标地址，建立连接及数据传送）、传输层（传输信息或报文）、会话层（协调、同步对话）、表示层（进行有关代码、字符、语法及其它方面的转换）及应用层（与用户的应用程序紧密相关）。既然它只是参考模型，所以多数 PLC 网络都不是完全按照这个模型进行构建的，而只是它的简化。

（1）企业网络，也称信息级、管理级，通信的数据量大，要求通信的速度高，但对通信的实时性要求可低些，允许短时间停止数据交换。

由于计算机网络目前用的几乎都是以太网，所以 PLC 与计算机联网最理想的也是用以太网。只是它要用在工业现场，所以所使用的组件要适用于工业现场。

OMRON PLC 的以太网有多种版本，如 10Base-5、10Base-T、100Base-TX 等。10Base-5 用的传输介质为同轴电缆，传输速率为 10Mbit/s，总线结构。10Base-T、100Base-TX 为非屏蔽双绞线，传输速率分别 10Mbit/s、100Mbit/s，星形结构。后者的功能也比前者强，但

可与前者兼容。

此外，还有冗余配置的以太网。同一 PLC 可配置两个以太网，一个工作，一个备份。一旦前者出现故障，后者将自动地无缝接替，用于可冗余配置的 CS1D 型 PLC。

冗余配置以太网的传输介质也是双绞线，速率也是 100Mbit/s。其功能除了冗余，其它与非冗余的基本相同。冗余以太网使用的模块也可配置成没有冗余的，故可与没有冗余的兼容。100Base-TX 版本新，不仅联网方便、成本低、与旧版本兼容；而且通信速率为原来的 10 倍。还新增了 E-mail 及 Socket 等原来所没有的功能。所以，是 OMRON 公司目前主推的以太网方案。最新推出的还有 CS1W-EIP21 和 CJ1W-EIP21 以太网模块，也可支持数据链接通信（说明见后）。通信速率及能力比过去的高出多倍，支持的通信协议也有增多。

组成以太网的主要是以太网模块或 PLC CPU 上集成的以太网接口（如 CJ1M CPU1 *-ENT、CJ2H-CPU68-EIP），此外，还有计算机的以太网卡及用于联网的集线器、双绞线电缆及附件。不过后者为以太网兼容产品，可按 OMRON 公司要求，从市场上选购。

当然，不是 OMRON 公司的所有 PLC 可接入以太网，只是配置有以太网模块或集成有以太网接口的机型才可接入。OMRON 以太网可以是简单的局域网，也可为互联网，还可冗余配置。

图 7-8 所示为简单的局域网实例。它有一台计算机及两台 PLC。前者用以太网卡，后者用以太网模块，用双绞线，通过集线器相连。联网后可实现计算机分别与两台 PLC 通信，也可实现两台 PLC 间通信。

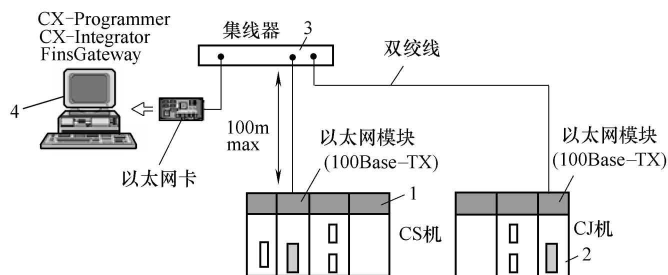


图 7-8　简单以太网配置

1—CS1W-ETN21　2—CJ1W-ETN21　3—集线器　4—计算机

图 7-9 所示为 PLC 以太网与企业内部网（Intranet）相接，同时企业内部网还通过防火墙、IP 路由器、服务器接入互联网。在这个 PLC 以太网上，计算机与 PLC 可以相互通信，与企业内部网络上的其它站点也可通信。只要防火墙允许，还可通过互联网与企业外部的计算机通信。

图 7-10 所示为冗余配置的 PLC 以太网。PLC 用的为 CS1D 型 PLC，有两个以太网模块。计算机也有两个以太网模块，分别通过 Hub（集线器）相接。这里一个为主，另一个为备份，可确保通信万无一失。

OMRON PLC 与计算机联网通信，除了以太网，还可用 Control-Link 网，Sysmac-Link 网，Sysmac-Net-Link 网。这些网的组建，在计算机上都要插入 OMRON 公司的专用网卡。在 PLC 方，要配置相应的网络模块，并要按要求选用联网介质。这些网络都是工业级的，可靠性高，但建网的成本也高，所以仅与计算机联网通信，一般不使用它。

（2）车间网络，也称为控制级、单元级，主要用于 PLC 间联网通信，以实现对多台设备或生产线的协调控制。它交换数据量小些，但对通信的可靠性、实时性要求很高，短时间停止数据交换是不允许的。

OMRON 公司中、大型 PLC 之间联网通信可使用的网络有 Controller Link 网、Sysmac-Link 网、PLC 链接网、Sysmac-Net-Link 网及以太网。而 OMRON 公司中、大型 PLC 与小型

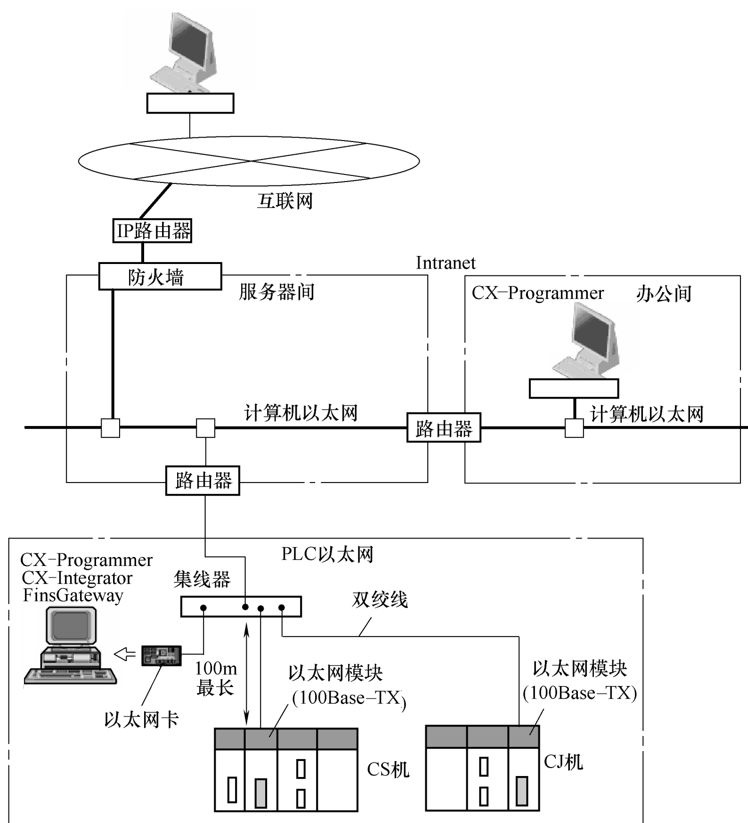


图 7-9　互连以太网

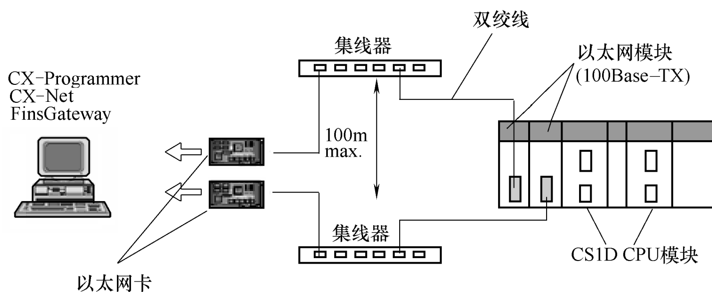


图 7-10　冗余配置 PLC 以太网

PLC 之间联网通信可使用的网络有、CompoBus/D 网（即 DeviceNet）、CompoBus/S 网、PLC I/O 链接网。较常用的只是 Controller Link 网及 DeviceNet。

Controller Link 网是 OMRON 公司于 20 世纪 90 年代后期开发的 PLC 控制网，用于 OMRON 公司中、大型机及高档小型 PLC，如 CP1H。如果计算机配置有 Controller Link 网卡，也可与计算机联网。

Controller Link 网所用的模块型号很多，通信介质可以是双绞线，也可为光纤，还可建冗余网络。较常用的是为 CLK 21 版本。它用双绞线为传输介质，可用重复器增大距离。还可通过光电转换器转换为使用光纤作传输介质，以增大传输距离。

图 7-11 所示为双绞线 Controller Link 网的简单连接。图示的计算机插上 Controller Link 网卡也可接入，也可与 PLC 一样进行数据链接通信。

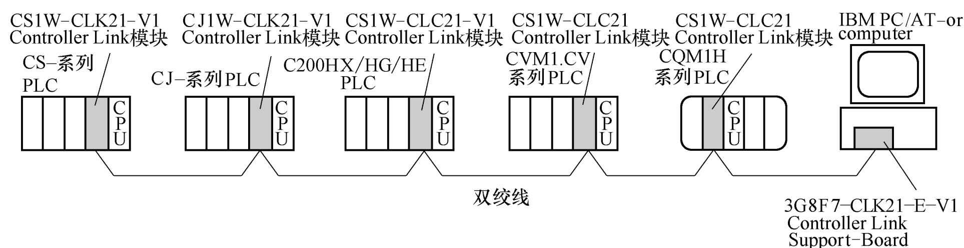


图 7-11　Controller Link 网连接示意

Controller Link 网站点数一般为 32 个，也可扩展为 64 个，通信速率可选定，与距离及传输介质有关，如使用屏蔽双绞线为传输介质，500m 时为 2Mbit/s，1km 时为 500bit/s。而且，它的通信方法多。交换的数据量大，一次通信可提交几 k 的数据。

Controller Link 网还可互联。图 7-12 所示为 Controller Link 网互联示意。

其中使用 2 个 Controller Link 模块的 PLC 作为 2 个网络的网桥，实现 2 个网络相连。再使用网络管理软件设定各个站点网络路径表，不同网络站点之间也可通信。此外，Controller Link 网还可通过网桥与以太网相连。

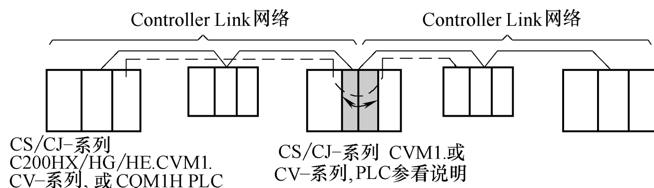


图 7-12　Controller Link 网互联示意

DeviceNet 是主从网。在 PLC 间联网主要是用在大、中型 PLC（设为主站）与小型 PLC（设为从站）间通信。可取代 OMRON 公司原有的 PLC I/O 链接网。

（3）现场网络。它主要用于 PLC 与现场设备及传感器/执行器通信，以实现 PLC 对现场设备及智能装置的信息采集与工作控制，交换数据量更小。但对通信的可靠性、实时性要求更高。即使短时间停止数据交换也是不允许的。

OMRON PLC 这类网络有 CompoBus/D 网（符合 DeviceNet 网标准）、CompoBus/S 网、PLC I/O 链接网。推荐使用的为 DeviceNet。

DeviceNet 是由 Allen-Bradley 公司（Rockwell 自动化公司）开发的一种基于 CAN（控制器局域网）的开放的现场总线标准。现在已经有超过 300 家的公司注册成为 ODVA（开放式 DeviceNet 供货商协会）的成员。全世界共有超过 500 家的公司提供 DeviceNet 产品。

DeviceNet 协议最基本的功能是在设备及其相应的控制器之间进行数据交换。DeviceNet 既可以工作在主从模式，也可以工作在多主模式。

DeviceNet 最大可以接入 64 个站点，可用的通信比特率分别为 125kbit/s、250kbit/s 和 500kbit/s 三种。设备可由 DeviceNet 总线供电（最大电流为 8A）或使用独立电源供电。DeviceNet 使用的电缆有粗电缆、细电缆和扁平电缆。

DeviceNet 设备的物理接口可在系统运行时连接到网络或从网络上断开，并具有极性反接保护功能。可通过同一个网络，在处理数据交换的同时对 DeviceNet 设备进行配置和参数设置。

7.1.3 PLC联网通信方法

PLC通信方法与通信平台有关。同一平台有多种通信方法。同一通信方法也可用于不同的平台。

1. 地址映射通信

图7-13所示为DeviceNet网络。主站用高档PLC，从站用低档PLC或智能设备。用于主从网络中站点之间通信。

图7-13中1为主PLC上的远程主控单元或接口，此为通信主站。而2、6为从PLC。3、7分别为从PLC 2、6上的PLC I/O链接或从站单元，此为通信从站。图中的4、5、8为主PLC的远程I/O终端，无自身CPU，分别作为主PLC的I/O点，受主PLC管理。

远程I/O终端与主PLC虽有通信，但它不是独立站点。作为主PLC的I/O点，在逻辑上与主PLC当地的I/O点没有区别，所差的只是I/O的响应时间略有通信所需的若干毫秒的延迟。

从站PLC有自身的CPU，还有自身的I/O，对其自身I/O的控制完全由自身管理。要与主站PLC交换信息，必须通过交换数据实现，关键是使用了PLC I/O链接或从站单元。它既是从PLC的扩展模块，在从PLC中有其I/O地址；又是主PLC的远程I/O终端，在主PLC中也有其映射的I/O地址，如图7-14所示。

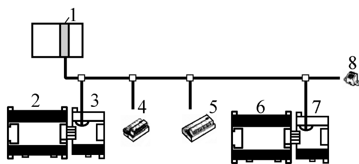


图7-13 主从网络

1—主PLC上的远程主控单元 2、6—从PLC
3、7—PLC I/O链接单元 4—网络终端器
5、8—主PLC远程I/O终端

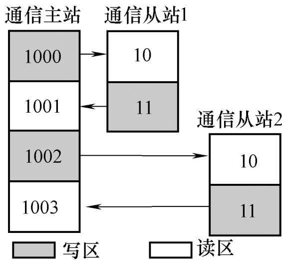


图7-14 地址映射通信机理示意

以图7-14所示为例，从站1的读（输入）区（字）10，在主站上的映射地址为1000，从站2的读（输入）区（字）10（不一定非是10），在主站上的映射地址为1002，且都是主站的写（输出）区（字）。从站1的写（输出）区（字）11，在主站上的映射地址为1001，从站2的写（输出）区（字）11（不一定非是11），在主站上的映射地址为1003，且都是主站的读（输入）区（字）。由此可知，每个从站在主站中都有专用于通信的数据读写映射地址。

这样，当主站轮流与各个从站及远程I/O终端通信时，将把主站映射区的数据依次发送给对应的从站输入区，各从站输出区的数据也依次返回给主站对应的映射区，而从站之间、主站远程I/O终端之间、从站与主站远程I/O终端之间不能通信，不能交换数据。

在这样系统中，如果从站1的PLC要向主站PLC发送数据，其具体过程可分为5步：

（1）把要向主PLC传送的数据，写入通信用输出区（如从站1为内存字11）；

（2）通过从 PLC 输出刷新，把数据传到 PLC I/O 链接单元或从站单元的存储区（如从站 1，为字 11 的地址）；

（3）通过网络通信，把 PLC I/O 链接单元或从站单元存储区的数据，送主站主控单元的存储区（字 1001 的映射区）；

（4）通过主 PLC 输入刷新，主站主控单元存储区的数据被读入主 PLC 的地址映射区（主 PLC 的字 1001）；

（5）主 PLC 从地址映射区（主 PLC 的字 1001）读取这个数据。

主 PLC 向从 PLC 发送数据的过程与此过程相反。先是主 PLC 向映射区写数据；再经主 PLC 输出刷新，传入主站主控单元的存储区；再通过网络通信，传入从 PLC I/O 链接单元或从站单元存储区；再经从 PLC 输入刷新，传到从 PLC 的输入存储区；最后由从 PLC 读取这个数据。

这里说的过程也很复杂，但它的输出、输入刷新都是 PLC 操作系统自动实现的。PLC I/O 链接单元或从站单元与主 PLC 主控单元间的数据传送，是由远程网络通信系统自动完成的，而且其通信过程如同 PLC 的扫描过程一样，总是周而复始地重复着，也非常可靠。

所以，对于这种通信，用户所要做的只是编写有关的数据读写程序，较为简单。只是，它所交换的数据量不大，多只有一对输入、输出通道（字），故只能用于较底层的网络上。

采用地址映射通信，输出对输入的响应也是有延时的，情况大体与链接通信类似，具体的不再赘述。

提示：用数据区地址映射通信只能在主站与从站之间进行。而从站之间通信则要通过主站转达。

提示：主站 PLC 及各从站 PLC“写区”、“读区”的大小取决于使用什么样的网络平台及作什么样的设置（组态）。

提示：主站除了为 PLC，也可为计算机，从站除了 PLC，也可能是智能设备，只要配置有相应的通信硬件就可以。

2. 数据链接通信

它用于 PLC 网络或有的串行接口联网通信。通信各方，指定相同或相对应的数据区地址参与数据链接。各站点都用广播方式发送数据，同时被其它站点接收。哪个站点成为发送站点，由“令牌”授权。谁拥有“令牌”，谁就成为发送站点。而“令牌”则总是在站点间轮流传递。具体管理由设定为主站的站点负责。其通信机理如图 7-15 所示。

图 7-15 所示为 4 个 PLC 进行数据链接通信的示意。字地址 1000～1063 之间的 64 个字被设置作为这个链接的数据区。而每个 PLC 都把它的这个区分为写与读两个部分，且这个划分对每个参与链接的 PLC 都是互补的。如 PLC1，其“写区”为 1000～1015 字，“读区”为 1016～1063 字，则 PLC2 的“写区”为 1016～1031 字，“读区”为 1000～1015 字及1032～1063 字，等等。

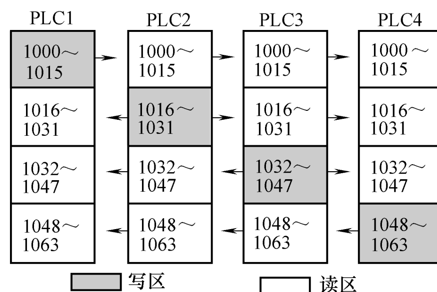


图 7-15 数据链接通信机理示意

这 4 个 PLC 的数据链接通信分 5 步实现：

（1）运行程序，把要向外传送的数据写入自身的写数据区；

（2）输出刷新，把“写区”数据传到与 PLC 链接单元或接口对应的（写）缓冲区；

（3）参与 PLC 链接的 PLC 链接单元或接口间相互传送数据，把与各链接单元或接口对应（写）的缓冲区数据传给与其它 PLC 的 PLC 链接单元或接口对应（读）的缓冲区；

（4）输入刷新，把 PLC 链接单元或接口缓冲（读）区的数据读入到读数据区；

（5）运行程序，把要使用的数据从自身的读数据区读走。

这里的输出、输入刷新是在程序扫描开始前或结束后由系统自动实现的。必要时，也可在程序中加入 I/O 刷新指令实现，以加快程序对链接数据的响应速度。

这里的各个 PLC 链接单元或接口间的数据传送，则由 PLC 链接系统的网络通信自动完成。其通信过程如同 PLC 的扫描过程一样，总是周而复始地重复着，力求使各个 PLC 链接单元的存储区的数据保持一致。

说得通俗一点，这里设置的链接数据区相当于邮局的信箱。数据发送与接收如同发信与收信，要外送的数据先投（写）入信箱，靠信箱的传送机制，把数据传送给对方；要用的数据则从别的 PLC 数据的信箱中取出。其间的数据传送如同邮局为你服务一样，是会自动实现的。若要快速传送，可另作 I/O 刷新，这如同寄快信一样。

总之，这个通信数据交换，经历了两种过程：

一是 PLC 的内存区中“链接区”与 PLC 链接单元或接口的缓冲区之间数据交换。这是由 I/O 刷新实现的。其周期取决于各 PLC 的程序扫描周期或程序中使用 I/O 刷新指令的情况。

二是各 PLC 链接单元或接口间的数据交换。它由主站 PLC 管理“令牌”，进而使各 PLC 轮流把链接单元或接口缓冲区中“写区”的数据，传送给其它 PLC 的“读区”。

这两个过程按照各自的周期重复进行着。也正是有了这两个过程，PLC 间的数据交换才成为可能，而且非常可靠。

链接通信交换的数据量比较大，可以为几个、几十个字。是很常用、很方便的 PLC 间的对等通信方法。具体大小取决于链接数据区的大小及参与链接的 PLC 数量。

提示：各个 PLC 的“写区”、“读区”的大小不一定都要相等。有的甚至可以只读或只写。这取决于使用什么样的网络平台及作什么样的设置（组态）。

提示：在链接通信之前，要对 PLC 作相应设定。这个设定有的由编程软件进行，有的用 PLC 执行初始化程序实现。

提示：参与链接的不一定都是 PLC，只要配置有相应的通信硬件，如计算机上安装有 OMRON Control Link 网络通信板及相关驱动程序，也可与联网的 PLC 链接通信。

3. 自由协议通信

在串口联网的平台上，双方根据自行约定，调用串行接口通信指令进行的通信称为自由协议通信，也称为无协议通信。

自由协议通信用于 PLC 之间通信，双方都要有串行接口通信指令。用于 PLC 与计算机，双方都要编程，主要是串行接口读、串行接口写，有的还有串口设定及使能。后两者用于串行接口初始化。当然这个设定与使能有的也可用编程软件完成。

要弄清的是，执行读、写指令只是对 PLC 串行接口缓冲区的读、写，而数据发送及接收则是由 PLC 通信管理系统分时完成的。一般讲，PLC 用于处理通信的时间大体占其程序

扫描时间的 10% ~20%。所以，执行数据发送指令后不是马上就把数据发送出去的。反之，不执行接收指令也不等于 PLC 不接收数据。只是把收到的数据放入接收缓冲区。

提示：只要串行接口的物理、电气、功能特性相同，双方又同处于可使用自由协议通信，那么使用各自的串行接口指令，还可以在不同品牌 PLC 间进行通信。

4. 串行接口协议通信

也是用于串行接口联网通信。主动方根据被动方的通信协议，向被动方发送通信命令；被动方接收到命令后，自行处理通信命令，并按要求向主动方应答。

显然，这里弄清通信协议是关键，否则主动方无法编程。在串行接口联网平台上，计算机与 PLC 通信、PLC 与智能设备通信多采用协议通信。

5. 网络协议通信

在 PLC 网络或有的串行接口联网的平台上，主动方（PLC 或计算机）按网络协议，执行网络通信指令，与被动方 PLC 通信。它交换数据量大，可达几百个字节；通信速度快（取决于网络的底层特性），而且只要有网关，有的还可跨网络通信，即网络中继运作（Network Relay Operations）。

网络通信指令有读、写及操作命令。主动方发送命令，被动方不须执行任何指令，都可对指定网络、指定站址、指定数据区进行读或写；可向对方发送操作命令，使对方改变工作模式；可强制或复位对方是工作位及文件操作等。

但是，被动方也可执行保护自身的指令。执行它后，可不让对方读、写自身数据，或受其操作。

网络协议通信要建立在 PLC 网络的平台上，而这样平台各厂商 PLC 多互不兼容，因此用这样通信只能在同类型 PLC 间进行。

6. 工具软件通信

用于计算机与 PLC 间联网通信。多可在多种通信平台上使用。

最常用为编程软件。用它可下载、上载程序与数据，控制 PLC 工作。

还有一些 PLC 厂商提供有监控软件或工具软件，也可与 PLC 通信。有了它，用户程序再与其做动态数据交换，即可间接与 PLC 通信。

再就是用厂商提供的通信接口函数或控件也可实现计算机与 PLC 通信。与通信工具软件不同的是，这些通信函数或控件不是单独的计算机程序，它要嵌入到用户的程序中。用户程序通过调用这些函数或控件去实现通信。

工具软件通信的好处是不必弄清 PLC 的通信协议，但要在计算机上装载这些软件、函数或控件。

7. 套接字（Socket）服务通信

PLC 以太网通信要用到套接字技术。套接字原是计算机应用程序使用 TCP（传输控制协议）和 UDP（用户数据报协议）的接口技术。而 OMRON PLC 的以太网也支持这个技术，以实现 PLC 与计算机以及 PLC 与 PLC 间的通信，而且在使用这种技术时，要预先做好 PLC 的设定，然后设法把 Socket（套接字）激活或调用网络通信命令进行通信。

PLC 的套接字通信服务支持 UDP（无连接通信），也支持 TCP（有连接通信），也还可执行 FTP（文件传输协议），以便在计算机与 PLC 间进行文件传输。

8. 利用互联网通信

最成熟通信是电子邮件，可发送、接收报文，还可添加附件，通信数据量可大到以兆位计。

此外，有的 PLC 可内置 Web 网页，提供网址，作为服务器。计算机用户可使用浏览器阅读此网页，读取 PLC 数据，以至上、下载文件。

9. 利用公共网络通信

公共网络是指公用电话网与移动通信网。使用调制解调器，通过电话拨号连接，实现计算机与 PLC 通信，是早已实现的技术，在远程通信中使用已有很多案例。

再就是利用移动通信网，实现 PLC 与计算机通信，也已在油田、供热、水处理这样分散的远程系统中开始应用。

7.1.4　PLC 通信程序特点

PLC 通信程序与控制程序、数据采集程序相比有如下几个特点：

1. 交互性

通信是双方的需要，也是双方要处理的工作，所以通信程序总是分布的。一般讲，在通信的各方都要编写有相应的程序。

这些程序大体有 3 类：

数据准备程序，用以提供要发送的数据，以备对方使用；

对话程序，用在通信中进行必要的发令与应答；

数据使用程序，用于读取对方发送的数据，并加以使用。

这 3 类程序是成对的，在各方又是相对应的。如甲方从乙方要数据，则甲方要编写“要数据命令及数据使用”程序，而乙方则要编写“数据准备及命令回应”程序。反之也一样。

2. 相关性

PLC 通信的目的是为了交换数据，以至于进行相互控制。而交换数据的方法与所使用的网络及其通信协议有关。PLC 的网络很多，协议也很多，尽管各厂商也力图建立一些标准网络，并作了很多努力，如 Profibus、DeviceNet 及 CC-Link 网，但至今，随着品牌及机型、接口的不同，具体的网络与协议仍是有很大的区别。

所以，PLC 通信程序有很强的与通信网络及其协议相关性。通信程序必须按照对象的协议编写，否则所编的程序无法实现通信。

附带在此提及的是，本章介绍的仅仅是通信程序编写，并不太牵涉到网络本身，只要使用的方法相同，不管什么网络，程序的算法则是相同的。要仔细介绍 PLC 网络，如仅仅介绍，就可能需要多本专著。好在这些网络的通信特性、组成模块及系统组态，各 PLC 厂商的说明书都有详细介绍。在系统集成时，也可取得有关代理商的技术支持，所以本书不对其展开介绍。

3. 从属性

PLC 通信、交换数据不是目的，而是为了使用这个数据。数据使用只能在有关控制或数据处理程序中实现。至于数据准备之前的工作，如数据采集、处理，也只是程序其它部分要作的工作。

所以，PLC 通信程序往往只是 PLC 整个程序的一部分，具有从属性。编写这类程序一定要与编写 PLC 其它程序配合与协调，才能取得通信程序的效果。

这样，与通信有关的程序，特别是和计算机与通信有关的程序是相当大的。它与其它应用一起，有的简直就是很大的软件工程，而通信程序，则是这个大软件工程中的一个从属部分。

4. 安全性

通信可靠，不出现数据或命令传送错误是很重要的。数据出错，特别是关键的控制用的数据出错，将出现灾难性的严重后果，所以通信可靠是绝对必需的。

为了通信可靠，除了硬件要有保证外，在软件上，也可采取很多措施，如报文校验、冗余通信等。

此外，还有通信安全问题。网络开放是好的，为系统的使用提供了方便，但也带来不安全的因素。因为不是什么数据都可让任何人知道，也不是任何人都有权去修改有关数据，所以，通信编程时，就要考虑到数据安全、保密、写保护等问题。

PLC 数据安全管理，OMRON 公司有的 PLC 有数据访问禁止指令 I/OSP（187），执行该指令可禁止外设或 Sysmac Net Link 网，或 Sysmac Link 网，或 Host Link 网……对本 PLC 内存区的读或写，由此可在一定程度上保护数据的安全。

与其对应的为数据访问允许指令 I/ORS，执行后允许访问。只是，这两条指令目前仅 CV 及其后续机拥有。

7.2　PLC 与 PLC 联网通信编程

关键词：T×D、R×D、SEND、RECV、CMD、地址映射通信、地址链接通信、串口指令通信、网络指令通信

PLC 与 PLC 通信可使用 PLC 串行接口或网络接口。对应的通信方法主要有链接通信、用地址映射通信、用串行接口指令无协议通信、用串行接口命令协议通信、用网络指令通信、用套接字（Socket）服务通信。

7.2.1　PLC 与 PLC 数据链接通信编程

1. 数据链接配置

数据链接通信平台可以是 Controller Link 网等 PLC 网络，也可是串行接口网络。以下以串行接口网络实例说明这个链接编程。

本例为 4 个 PLC 链接，主（母）站为 CP1H CPU 单元（或 CJ1M CPU 单元），各从（子）站为 CP1H CPU 单元（或 CJ1M CPU 单元）。设定为全站链接方式。链接的通道（CH）数为 10 CH。其配置及设定如图 7-16 所示。

在实施链接通信之前，要用 CX-Programmer 编程软件进行相关的链接设定。

（1）主站 PLC 的设定。其设定如图 7-17 所示。

从图 7-17 可知，它用的是串行接口 1（如图中 1），设为主站（如图中 2），通信设定波特率为 9600bit/s，格式为“7，2，E”（如图中 6），链接字为 10CH（如图中 3），链接方（模）式为“全部”（如图中 4），NT/PC 链接最大数为 3（如图中 5）。

（2）从站 PLC 的设定。其设定如图 7-18 所示。

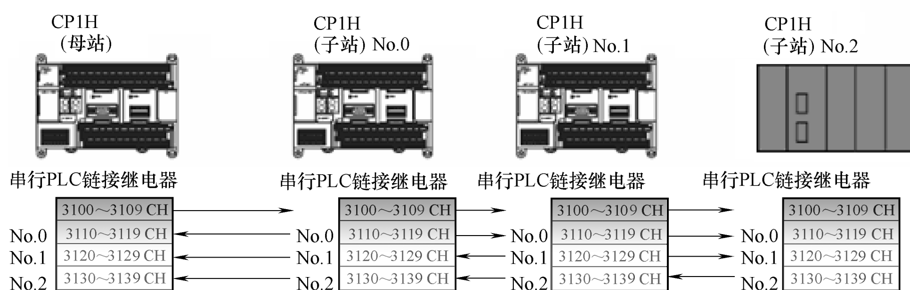


图 7-16　PLC 链接配置之一

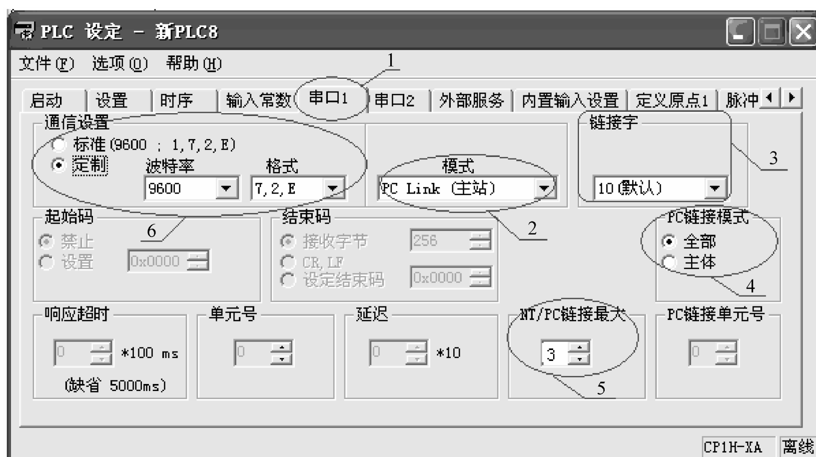


图 7-17　PLC 链接主站设定

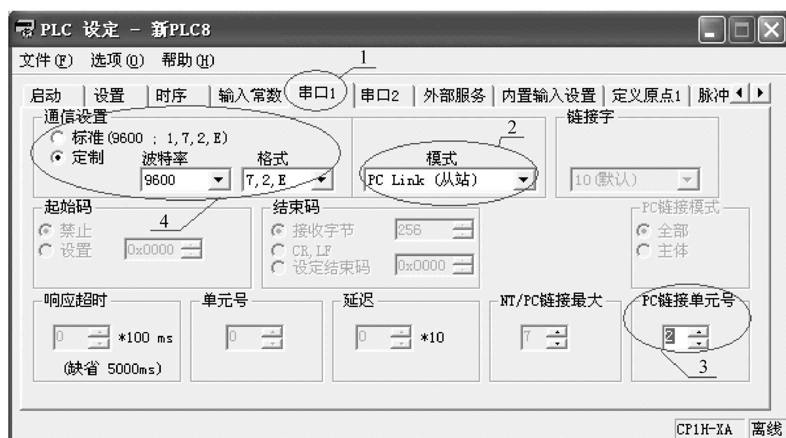


图 7-18　PLC 链接从站设定

从图 7-18 可知，它用的也是串口 1（如图中 1），设为从站（如图中 2），通信设定与主站相同（如图中 4），PC 链接单元号为 2（如图中 3）。其它从站也要作类似设定，具体略去。

上述设定的含义是，所使用的链接地址为 3100～3139。其中主站的“写区”为 3100～3109，其余的为“读区”。从站 0 的“写区”为 3110～3119，其余的为“读区”。余类推。

把这些设定下载给各 PLC，当进行通信时，各 PLC“写区”的内容将依次自动传送给各 PLC 相同地址的“读区”。这样，各 PLC 要向外传送的数据可写入自己的写区，而要使用别的 PLC 的数据可从别的 PLC 的写区读取，从而实现这几个 PLC 间的数据交换。

本例的 4 个 PLC 链接，也可设定为主站链接方式。链接的通道（CH）数为 10CH，如图 7-19 所示。

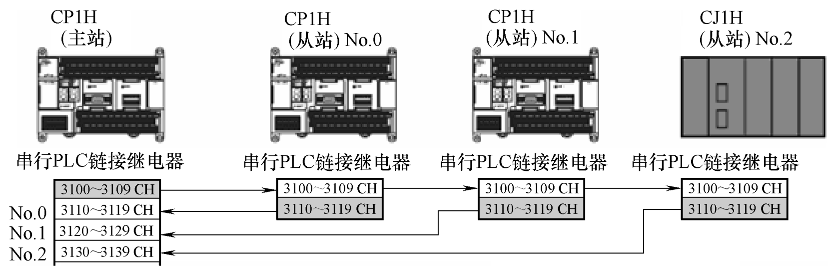


图 7-19 PLC 链接配置之二

其主站设定与图 7-17 不同的只是把链接模式改为“主体”，而从站设定不变。这样设定的含义是，主站所使用的链接地址为 3100～3139。其中主站的写区是 3100～3109，其余的为“读区”。而各子站的写区均为 3110～3119，而“读区”均为 3100～3109。只是在主站上，对各子站的“读区”不同。对从站 0 的“读区”为 3110～3119，对从站 1 的“读区”则为 3120～3129、对从站 2 的“读区”为 3130～3139。只是，这样链接从站间不能通信，但从站与通信相关的程序相同，便于程序共享。

提示：对 CP1H 机，在串行接口1、2中，只能有一个接口设定为“串行 PLC 链接主站”或“串行 PLC 链接从站”模式，如都设定为“串行 PLC 链接主站”或“串行 PLC 链接从站”，则出现 PLC 系统设定异常（运行继续异常）、A402 CH 位10（PLC 系统设定异常标志）为1（ON）。

提示：最早,OMRON PLC 只是 PLC 专用的网络才有数据连接通信功能。如今，由于技术进步，标准串行接口互连也有此功能。

2. 链接通信编程

图 7-20 所示为数据链接通信发送数据方程序。图中用的是符号地址。

从图 7-20 可知，这里①是把“工作数据”写入“PLC1 链接”写区“字 0”。其实，如不要求对方回应，整个通信程序本身仅此已足够。

图 7-20 所示的②、③、④指令调用及图 7-21 所示程序，是为了能得到对方的回应而增加的。

图 7-21 为 PLC2 的程序，它的作用是把读到的数据，返回给对方的“链接区字 16”。如系统通信畅通，PLC2 工作正常，则“链接区字 1”与“链接区字 16”肯定是相等的。那么，经图 7-20 指令②比较，不会出现不相等。从指令③、④知，不会产生“数据不一致报警”信号。反之，如系统通信不畅或 PLC2 工作不正常，那么，经 2s 延时，定会产生“数据不一致报警”信号。

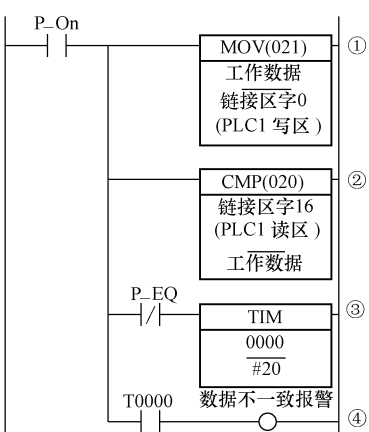


图 7-20 链接通信发送数据方程序

由此，发送数据方即可采取对应措施，以确保系统安全。

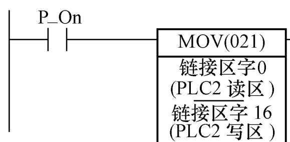


图 7-21　链接通信 PLC2 程序

但是，如这里“工作数据”较长时间不变，有可能检测不出通信或对方工作不正常。为此，在读数据方，最好也要认定所读的数据，是否是正确的程序。其方法是定时地把“读区”数据先转存给数据缓冲区，然后用非法数据改写“读区”。下一次再读时，如“读区”存的仍为非法数据，说明通信或对方工作不正常。图 7-22 所示即为这个程序。

从图 7-22 可知，①处可产生 2s 脉冲信号（T0001）。T0001 ON 将执行指令②、③、④。先看非法数据是否被链接通信所更新。如未更新，说明数据链接出错，则“读数错报警”ON，否则所读数据转存给数据缓冲区，并用非法数据再改写“读区”，为下一轮的检查做准备。

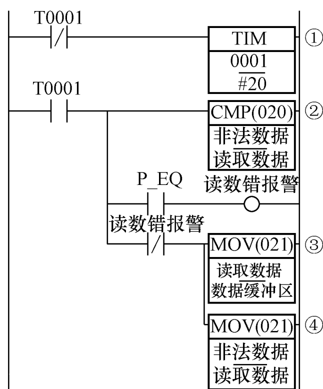


图 7-22　定时检查通信是否正常程序

提示：如果有响应通信状态的辅助继电器，可用以处理通信故障，虽不直接，但可能更简便些！

7.2.2　PLC 与 PLC 地址映射通信编程

1. 地址映射通信配置

图 7-23 所示为 DeviceNet 配置例子。该配置主站为 CJ1 型 PLC，使用的主单元为 CJ1W-DRM21，从站点有 3 个，使用的从单元为 CPM1A-DRT21。各个站点地址及有关参数可以在 CX-one 软件包中 CX-Integrator 集成软件上设定。模块上也有设定开关，也可以设定。当然，这两个设定必须一致。

这里，各个从站 CPM1A-DRT21 都有输入、输出各 2 个通道用于通信。其地址与其它 I/O 扩展单元一样，与所用的 CPU 与 I/O 配置有关。如#01 站点用 CPM1A CPU30（30 点），CPU 单元即连接 CPM1A-DRT21，那么它通信输入通道为 2、3，输出通道为 12、13。

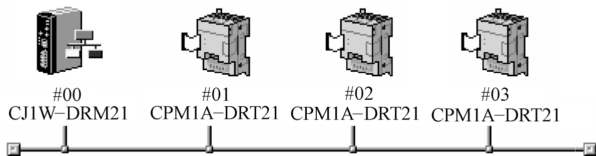


图 7-23　DeviceNet 网络配置例

从站#01、#02、#03CPM1A-DRT21 输入通道的地址，在主站#00 上的映射地址为：

#01　CPM1A-DRT21　4 Byte　3300：Bit00（起始位）

#02　CPM1A-DRT21　4 Byte　3302：Bit00（起始位）

#03　CPM1A-DRT21　4 Byte　3304：Bit00（起始位）

从站#01、#02、#03CPM1A-DRT21 输出通道的地址，在主站#00 上的映射地址为：

#01　CPM1A-DRT21　4 Byte　3200：Bit00（起始位）

#02　CPM1A-DRT21　4 Byte　3202：Bit00（起始位）

#03　CPM1A-DRT21　4 Byte　3204：Bit00（起始位）

这样配置的含义是，主站写地址 3300 通道，就等于写#01 从站的 2 通道。主站读地址 3200 通道，就等于读#01 从站的 12 通道。反之，从站#01 写 12 通道，就等于写主站的 3200 通道。从站#01 读 2 通道，就等于读主站的 3300 通道。

2. 地址映射通信编程

图 7-24 所示为用主站上的一个起停按钮，去控制从站的一个装置的主机程序。

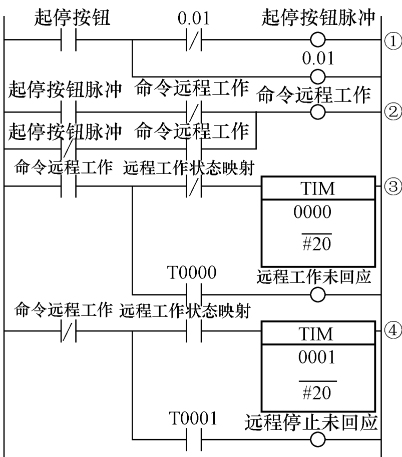


图 7-24 地址映射通信主机程序

图 7-24 中①把“起停按钮”ON 信号转换为“起停按钮脉冲”ON 一个扫描周期信号。②用以控制“命令远程工作”的起停（ON/OFF），这个电路已在第 2 章中做过讨论，这里不再重复。“命令远程工作”的状态将通过网络通信，复制给下位机的“命令远程工作映射”。而下位机的程序如图 7-25 所示，很简单，只有 2 个梯级指令，第一梯级指令的目的是把“命令远程工作映射”赋值给“远程工作”。

有了以上的对应程序，用主站上的一个起、停按钮，即可控制从站的一个装置工作。但仅此是不够的，发出工作命令后，还应弄清，控制命令是否执行了。

为此，图 7-24 中加入了③、④梯级程序。从图知，只要“命令远程工作”状态改变，就可使定时器工作。如“命令远程工作”与“远程工作状态映射”一致，则定时器工作停止。而“远程工作状态映射”是下位机“远程工作状态”的映射，是通过通信传来的。从图 7-25 梯级②知，“远程工作状态”是“远程工作”的直接赋值，故只要在一定时间（为了通信传送数据需要）内，下位机实现了上位机“命令远程工作”的要求，“远程工作未回应”或“远程停止未回应”都不会 ON，否则就可能 ON。依此，上位机就清楚了下位机是否执行了给其控制的命令了。

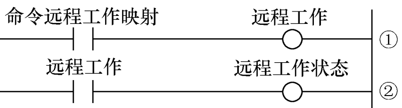


图 7-25 地址映射通信从机程序

7.2.3 PLC 与 PLC 自由协议通信编程

1. OMRON PLC 串行接口通信指令

自由协议通信双方都要用到串行接口通信指令。OMRON 串行接口指令较多。同样指令，机型不同，功能也不完全相同。表 7-3 所示为 CJ 系列 PLC 的串行接口指令。

表 7-3 CJ 系列机串口指令

指令语言	助记符	FUN 编号	指令语言	助记符	FUN 编号
协议宏	PMCR	260	串行端口输出	TXDU	256
串行端口输出	TXD	236	串行通信单元	RXDU	255
串行端口输入	RXD	235	串行端口输入		
串行通信单元	TXDU	256	串行端口通信设定变更	STUP	237

串行接口帧格式也有多种，如图 7-26 所示。使用什么帧格式，由调用指令时使用的控制字决定。

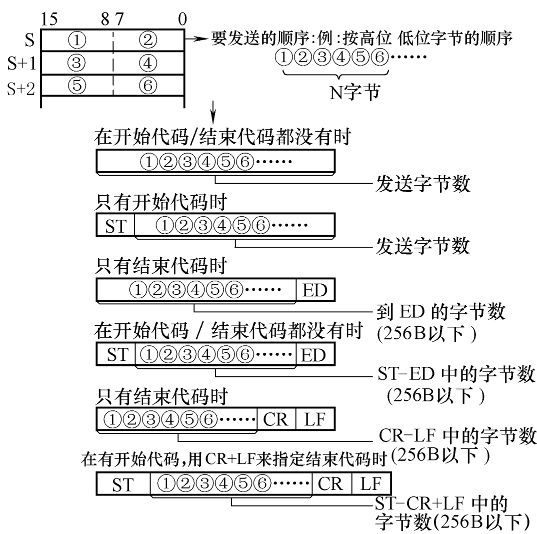
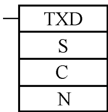


图 7-26　帧格式

（1）发送指令 TXD。用以向串行接口发送数据，指令的梯形图格式如下：



这里，S——指定发送数据存放的首地址；

C——控制字；

N——指定发送数据的字节数。

执行本指令，将从串行接口发送操作数 N 指定字节的数据。数据的开始地址由操作数 S 指定。选用哪个串行接口及通信帧格式由控制字 C 指定。控制字 C 的含义如图 7-27 所示。

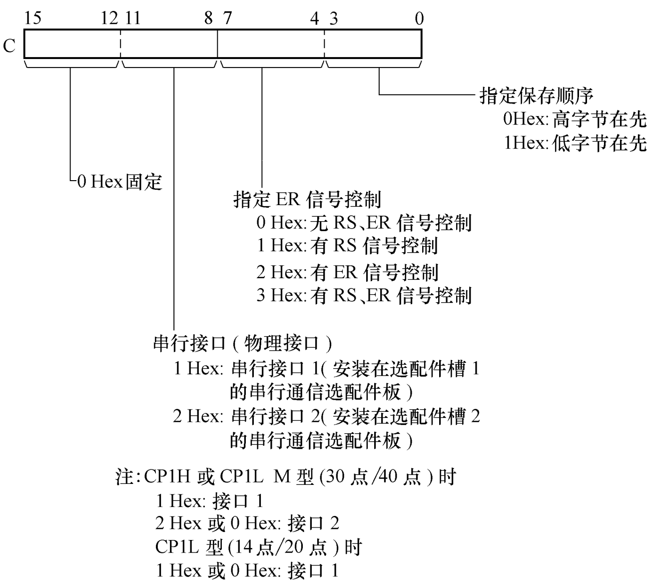
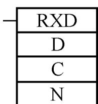


图 7-27　发送控制 C 字含义

不过，数据真正发送还需在串行接口发送准备就绪后才能实现。而这种就绪则是用辅助继电器 A392.13（对通信板 1）和 A392.05（对通信板 2）ON 表示。

最多可发送的字节数为 259B。但除去帧的开始及结尾，真正包含的数据最多为 256B。

（2）接收指令 RXD。用以从串行接口读取数据，指令的梯形图格式如下：



这里，D 指定接收数据存放的首地址，C 为控制字，N 为指定接收数据的字节数。

执行本指令，将从串行接口接收数据 N 指定字节的数据，并予以存储。数据存储的开始地址由操作数 D 指定。选用哪个串行接口及通信帧格式式由控制字 C 指定。控制字 C 的含义如图 7-28 所示。

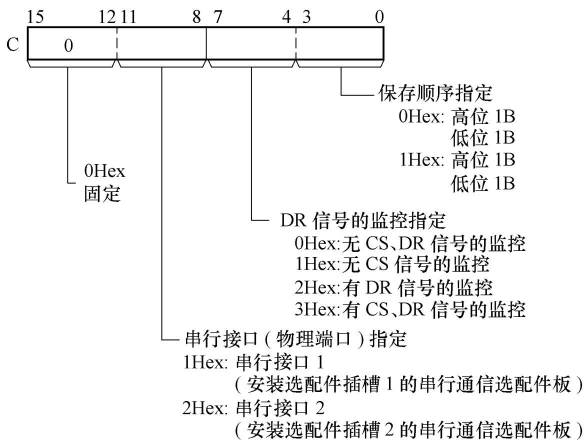


图 7-28　接收控制字含义

不过，数据接收还需在串行接口接收准备就绪后才能实现。而这一就绪则是用辅助继电器 A392.14（对通信板 1）和 A392.06（对通信板 2）ON 表示。

最多可接收的字节数为 259B。但除去帧的开始及结尾，真正包含的数据最多为 256B。

使用发送、接收指令，首先要做好串行接口设定，其次还要弄清有关通信控制与状态辅助继电器的含义及使用。表 7-4 所示为有关辅助继电器的功能及说明。

表 7-4　串行接口有关辅助继电器功能及说明

名　称	地　址	内　容
串行接口 1 接收完成标志	A392.14（CP1H、CP1LM）	在自由协议模式中接收结束时为 1（ON） 接收字节数指定时，接收指定字节数时 ON 结束代码指定时，在结束代码接收或 256B 接收中为 ON
	A392.06（CP1LL）	
串行接口 2 接收完成标志	A392.06	
串行接口 1 接收超限标志	A392.15（CP1H、CP1LM）	在自由协议模式中，超越接收数据量进行接收时为 1（ON） 接收字节数指定时，接收结束后在执行 RXD 指令之前，进行数据接收时为 ON 结束代码指定时，结束代码接收后在执行 RXD 指令之前，进行数据接收时为 ON 结束代码未接收时，256B 接收后，第 257 字节不为结束代码时为 ON
	A392.07（CP1LL）	
串行接口 2 接收超限标志	A392.07	
串行接口 1 接收计数器	A394CH（CP1H、CP1LM）	自由协议模式时，对接收数据的字节数用十六进制数来表示
	A393CH（CP1LL）	
串行接口 2 接收计数器	A393CH	
串行接口 1 端口再起动标志	A526.01（CP1H、CP1LM）	将这个标志设为 ON 时，对串行接口进行初始化接收结束标志和接收超限标志为 OFF，接收计数器为 0。还有接收缓冲器也被清除 处理结束后，这个标志在系统中变为 OFF
	A526.00（CP1LL）	
串行接口 2 接口再起动标志	A526.00	

提示：早期 OMRON C 系列机没有串行接口指令。串行接口只能按 Host Link 协议被动接受通信。自 C200Hα 型 PLC 后，才有串行接口指令，才可主动发起通信。在主动发起通信的同时，仍可使用 Host Link 协议被动通信。但对 CJ 型 PLC 来说，当处于 Host Link 协议时，不能调用通信指令，而处于 RS-232模式时，才可调用。

2. OMRON PLC 串行接口通信，一台发送数据，另一台接收数据程序实例

两台 PLC 通信，PLC1 定时发送数据，PLC2 随时接收。在通信前，首先要分别对两台 PLC 都做好串行接口设定。设定可使用 CX-Programmer 软件，通过 USB 接口与 PLC 联机后进行。图 7-29 所示为串行接口 1 设定的情况。

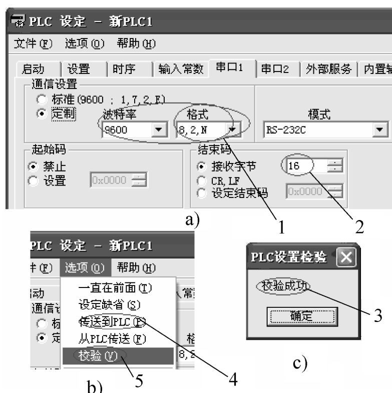


图 7-29 串行接口 1 设定
a）PLC 设定窗口 b）下拉菜单
c）PLC 设置检验窗口

联机后，首先要使 PLC 处于编程状态。接着打开设定窗口。这里选用通信板 1，即串行接口 1。从图 7-29 可知，通信设置为定制，即用户自定。图中 1，选波特率为 9600bit/s，格式为 8，2，N，表示为 8 位数据位（可保证传送所有二进制数据），2 位停止位，无奇偶校验。图中 2 所示的通信模式为 RS-232C，接收字节数为 16B，不用起始码及结束码。这样选择，所使用的数据区不超过 16B。

设定后，点击窗口上“选项”下拉菜单，再在其上，选“传送到 PLC”项（图中 4）。按提示操作，将把这里的设定下载给 PLC。

下载后，可再在“选项”下拉菜单中，选“校验”项，将进行是否下载成功的检查，如成功将弹出“PLC 设置检验”窗口，并提示“检验成功”，否则将显示“检验错误”。

提示：有时，设定下载后，须对 PLC 重新上电，设定才能生效，即检验才能成功。

除了设定，还要编程。图 7-30a 所示为 PLC1 为定时发送数据程序。而图 7-30b 所示为 PLC2 定时接收数据程序。

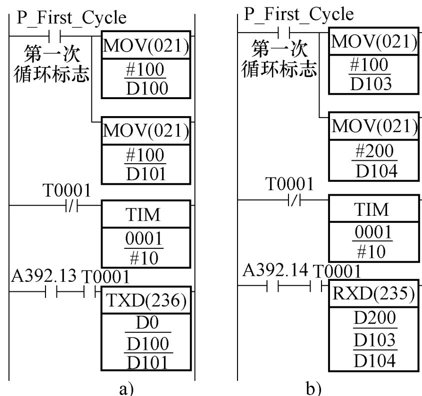


图 7-30 PLC 定时发送、接收数据程序
a）发送数据程序 b）接收数据程序

从图 7-30 可知，两个程序都是先进行初始化，做好通信准备。图 7-30a 把 D100（控制字）赋值为十六进制数 100（使用串行接口 1，无起始及结束字节，先发送高字节），D101 赋值为十六进制数 10（发送 16B、8 个字）。图 7-30b 把 D103（控制字）赋值为十六进制数 100（使用串口 1，无起始及结束字节，先发送高字节），把 D104 赋值为十六进制数 10（发送 16B、8 个字）。

进而，都是进行定时器调用，目的都是生成 1s 脉冲信号 T1。接着 PLC1 是在发送就绪标志 ON 及定时时间到时，调用发送指令。而 PLC2 是在接收完成标志 ON 及定时时间到时，调用接收指令。

可知，PLC1 执行图 7-30a 程序，将相隔 1s 钟都将把 D0 ~ D7 间 8 个字、16B 的当前值向 PLC2 发送。当然，不用定时发送，而用先设

置逻辑条件，在条件满足时发送也是可以的。

而 PLC2 执行图 7-30b 程序，将相隔 1s 将把从串行接口读取的数据，存入 D200 ~ D207 间 8 个字、16B 中。

由于这里用的是 RS-232C 接口，是全双工的，所以当两台 PLC 同时都运行既有发送指令又有接收指令的程序，那么即可做到，各 PLC 都可把自身 D0 ~ D7 的 8 个字的数据，写到对方的 D200 ~ D207 的 8 个字中，实现实时的数据相互交换。

一般讲，数据发送比数据接收要简单些。发送准备就绪，执行发送指令总能把数据发出。但数据接收则比较麻烦些。当串行接口出错、接收数据溢出等情况，将不能接收数据。有时，还可能出现数据存储“串位”。图 7-31 所示为改进的数据接收程序。

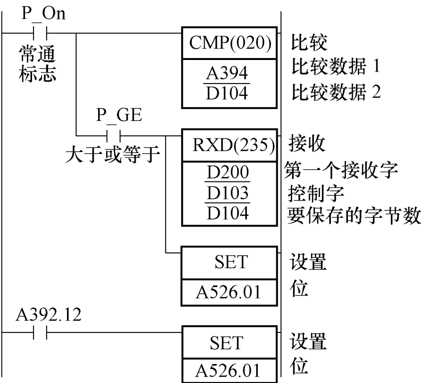


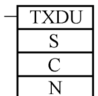
图 7-31 改进的数据接收程序

从图 7-31 可知，程序总是进行接收计数器与设定的接收数据字节设定值比较，一旦出现前者大、等于后者，将执行接收指令；之后还使串口复位、初始化，为新的接收作准备。而且，一旦串行接口出错，即标志 A392.12ON，也将初始化串行接口，经实际调试，效果更好。

3. OMRON PLC 串行接口通信模块通信指令

OMRON CJ1、CP1H 等机型还可配置串行接口通信模块。它不能使用上述串行接口指令，而要用自身的通信指令。

（1）发送指令。用以发送数据，其梯形图格式为



这里，S——指定发送数据存放的首地址；

C——控制字；

N——指定发送数据的字节数。

执行本指令，将从 S 到 S +（N/2）-1 源字中读取数据，在串行接口通信模块处于自由协议模式下，将所读数据向通信对方发送。选用哪个模块、模块上哪个通信口以及通信帧格式由控制字 C 及 C +1 指定。控制字 C 及 C+1 的含义如图 7-32 所示。

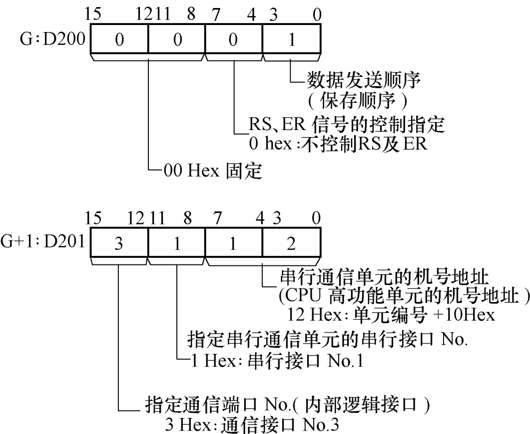


图 7-32 发送控制字含义

与 TxD 命令不同的是，它的控制字有两个：C 仅用于低字节，功能与 TxD 的相同；C +1 是新加的，其低字节用以指定通信模块的地址。通信模块的地址与所设定的机号有关，可按下式计算：

80(十六进制)+4×机号(对口1)

或　81(十六进制)+4×机号(对口2)

其高字节的第 8～11 位，指定使用模块上的串行接口 0（设定值为 0）或 1（设定值为 1）。高字节的第 12～15 位，指定通信使用的逻辑通信口，在 0～7 之间选择。因为通信模块通信要接收 FINS 协议管理。不同通信口可共用一个逻辑通信口，但同时只能一个使能(Enabled)。使能与否，分别由辅助继电器 A202.00～202.07 的状态位（对应逻辑口 0～7）显示。还有其它一些辅助继电器，其功能见表 7-5。

表 7-5　有关辅助继电器功能

名　称	地　址	内　容
网络通信指令可执行标志	A202.00～A202.07	网络通信(包括 TXDU 指令)在能够执行时为 1(ON) 各位对应通信端口(内部逻辑端口)号为 00～07,通信端口 No.00～07 在网络执行中为 0(OFF),在执行结束(不管是正常还是异常)后为 1(ON)
网络通信响应代码	A203～A210CH	网络通信被执行时保存响应代码(结束代码) 各 CH 对应通信端口(内部逻辑端口)No A203～A210CH:通信端口号:00～07 在通信指令执行中为 00Hex,反映在通信指令执行结束时运行开始时清除
网络通信执行出错标志	A219.00～A219.07	在网络通信执行中发生出错(异常)时为 1(ON) 各位对应通信端口(内部逻辑端口)号:00～07,通信端口 No.00～07 在执行下一个网络通信之前,状态被保持。即使异常结束,也要到在下一个的通信指令执行时为 0(OFF)

此外，使用串信接口通信模块通信所作的设定不像 CPU 模块上的串信接口，使用 CX-Programmer 软件设定窗口进行设定。而用 I/O 表及其设置项进行。也可对指定的相关数据区赋值实现。相关数据区与设定模块机号有关。其计算公式如下：

m = D30000 + 100 × 单元编号 CH

n = 1500 + 25 × 单元编号

而对 CP1H 机，其计算公式为

数据存储区：m = D30000 + 100 × 模块机号

核心 I/O 区：n = CIO 1500 + 25 × 模块机号

而这些内存区的设定及状态如表 7-6、7-7 所示。

表 7-6　有关数据存储区的设定及状态

名　称		位	内　容	设　定
端口 1	端口 2			
m+2	m+12	15	自由协议模式时发送延迟时间	0:默认值(0ms) 1:任意设定
		14～00	自由协议模式时发送延迟任意设定时间	(0000～7530hex:十进制 0～300,000ms)[10ms 单位]

（续）

名　称		位	内　容	设　定
端口 1	端口 2			
m+4	m+14	15～08	自由协议模式时开始代码	00～FF hex
		07～00	自由协议模式时结束代码	00～FF hex
m+5	m+15	15～12	自由协议模式时开始代码有无设定	0hex：无 1hex：有
		11～08	自由协议模式时结束代码有无设定	0hex：无 1hex：有 2hex：CR+LF 指定

表 7-7　有关内存位状态

通　道		位	内　容
端口 1	端口 2		
n+9	n+19	05	TXDU 指令执行中标志：1：执行中、0：非执行中

图 7-33 所示为这个指令的使用实例。

本例的使用模块的机号设定为 2；用端口 1；所使用数据区为 D30205 为 1100hex，含义为使用帧起始、结束代码；D30204 为 203hex，含义是制定帧起始代码为 02hex（ST）、结束代码为 03hex，D30205 设定为 003hex。

控制字 C 设为 1hex，C+1 为 3112hex。含义为低字节先发送，指定逻辑端口 3，使用串行接口 1 及串行接口通信单元地址为 12hex。

结合本例，如果原数据区的内容如图 7-34a 所示，那么，对应所发送的字符串将是如图 7-34b 所示。

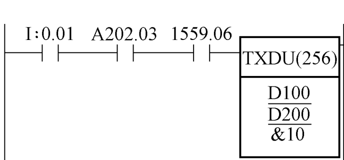


图 7-33　使用发送指令程序

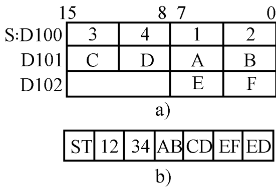
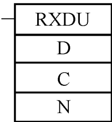


图 7-34　原数据区内容

执行本程序，只要逻辑通信接口 3 使能（A202.03ON）、指令又不是执行中（1559.05OFF），当 0.0 ON 时，将把字符串“ST”+“1234ABCDEF”+“ED”发送给通信对方。

（2）接收指令。其梯形图格式为：



执行本指令，将从通信模块的串口上，接收通信对方发送的、由操作数 N 指定字节的数据。所接收数据存储开始地址由操作数 D 指定。而选用从哪个模块、模块上哪个通信口接收，以及通信帧格式由控制字 C 及 C+1 指定。控制字 C 及 C+1 的含义如图 7-35 所示。

与 RxD 命令不同的是，它的控制字有两个：C 仅用低字节，功能与 RxD 的相同；C+1 是新加的，其低字节用以指定通信模块的地址，通信模块的地址使用及相关设置与 TXDU 指令相同。

提示：CJ 系列机有关串行接口通信模块有多种型号与版本。不是所有机型都支持这两个指令。是否支持，在使用之前可用 CX-Programmer 软件对其进行配置，如可设置成自由协议通信模式，则支持；否则不支持。

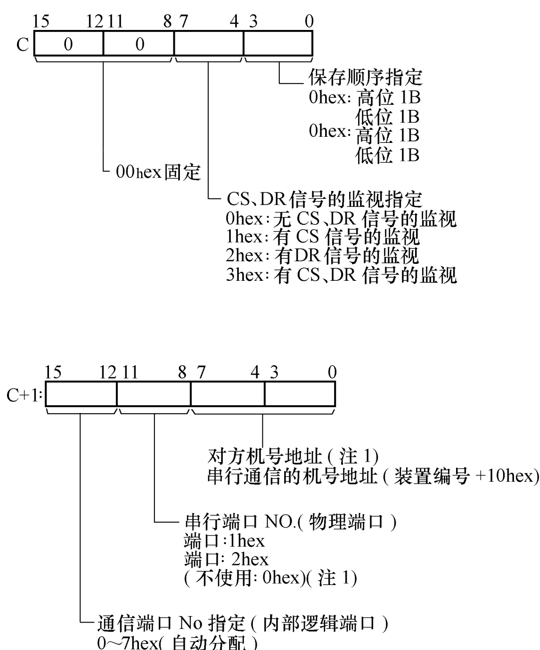


图 7-35　控制字设置

注：在串行接口 No（物理端口）中指定 00hex，在对方机号地址中能够直接指定串行接口的机号地址（串行接口 1：80hex +04hex × 单元编号
串行端口 2：80hex +04hex × 单元编号）

7.2.4　PLC 与 PLC 串行接口协议通信编程

在串行接口联网的平台上，通信主动方 PLC 处于自由协议模式，根据通信被动方 PLC 厂商通信协议，调用有关串行接口通信指令或函数实现与被动方 PLC 通信。

与自由协议通信不同的是，协议通信只需在主动方执行发送通信命令的程序，而被动方无须执行通信程序。协议通信另一好处是，被动方可以是低档的、不具备串行接口通信指令的 PLC，如 OMRON CPM1A 型 PLC。

有的 PLC，如 OMRON CPM2A 型 PLC，处于协议模式下也可执行通信指令，那通信的主动方也可处于协议模式下通信。

用协议通信虽较为简单，但要弄清 PLC 的通信协议，否则这样的通信是无法实现的。

PLC1 为通信主动方，用自由协议，设定为 RS-232 模式，要执行发送与必要的接收程序。PLC2 为被动方，用 Host Link 协议，设定为 Host Link 模式，不用执行程序。

图 7-36 所示为协议通信准备程序。而调用发送指令的细节与自由协议通信相同，在此不再重复。该程序的通信命令为 ASCII 字符串，本例为“@00WR0100FFFF”。其含义是使对方 PLC 的 0100 通道的各个位置 1。

提示：要使这个通信命令实现，还要预先把对方 PLC 处于“监控”模式。如果处于“运行模式，则是无法改写 PLC 数据的。

提示：有关 Host Link 协议及 FCS（校验码）见本章7.2.5节介绍。

该程序运行开始时，即把命令用的 ASCII 字符的十六进制值，用双字传送指令赋值给

D100 到 D107 中（D107 仅用高字节）。接着调用 FCS 功能块，以进行校验码计算。这是 Host Link 协议要求的。校验码是上述命令字符 ASCII 码十六进制制值的异或而得的。占两个字节。不足两个字节时，高位应补 0，加在上述命令码之后。

计算 FCS 校验码功能块用 ST（结构化文本语言）编写。其变量设置如图 7-37 所示。这里有输入变量、输出变量及内部变量。

其中图 7-37a 所示内部变量定义为字数组变量，而且用 AT 地址，即与实际内存地址关联，所以，图 7-37a 所示数组虽为内部变量，但实际为实际内存地址，因此，实质上是用作通信命令字符串的输入。图 7-37c 所示输入变量 dNum，为无符号整型数 INT，指明通信命令字节数。图 7-37b 所示输出变量为字 FCS，存放校验码 ASCII 字符的十六进制值。

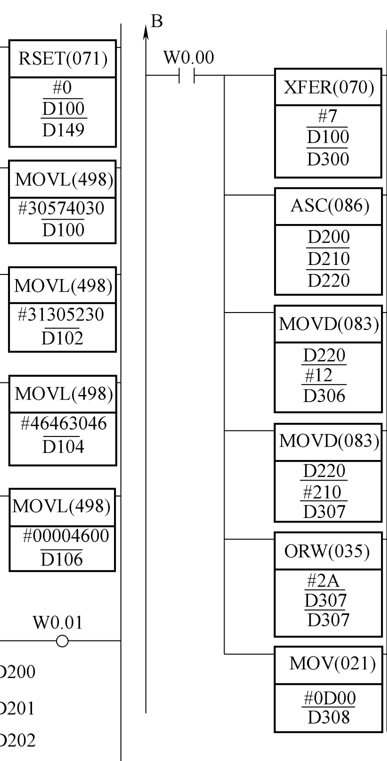


图 7-36　协议通信准备程序

图 7-37a 所示数组设定如图 7-38 所示。图中“编辑变量”窗口是功能块在选择插入变量时打开的。

名称	数据类型	AT
I1	INT	
TeH	WORD	
TeL	WORD	
TeW	WORD	
TeHi	INT	
a	WORD [50]	D100
TeLi	INT	
dL	INT	
TeWi	INT	

内部　输入

a)

名称	数据类型	AT
ENO	BOOL	
FCS	WORD	
tt	WORD	
ATadr	WORD	

内部　输入　输出

b)

名称	数据类型	AT
EN	BOOL	
dAdrr	WORD	
dNum	INT	

内部　输入

c)

图 7-37　变量设置

a）内部变量　b）输出变量　c）输入变量



图 7-38　编辑变量及高级设置窗口

从图 7-38 可看出，在打开的“编辑变量”窗口上，有“高级”按钮项（只有内部变量编辑时此按钮才激活）。单击此“高级”按钮，将弹出“高级设置”窗口。之后，可在“高级设置”窗口上，选择“数组变量”，并定义数组大小（本例选用 50）；再选择“AT（指定地址）”，规定的可在为 D0～D32718 范围内选择（本例选 D100）。选择后，单击“确

定”按钮，高级设置成功，并关闭“高级设置”窗口。再在“变量设置”窗口上，单击“确定”按钮，数组设定成功，并关闭“变量设置”窗口。

经过这样设置，数组下标 0，将指向 D100；下标 1，将指向 D101。余类推。建立与实际地址关联的数组，为计算 FCS 提供了方便。

而用 ST 语言编写的计算 FCS 校验码的程序代码如下（代码含义见注释）：

```
TeW:=0;(*先把结果初值设为0程序*)
FOR I1:=0 TO dNum DO(*自 D100 开始,逐字循环进行异或计算,结果存入 TeW 中*)
   TeW:=a[I1] xor TeW ;
END_FOR;
TeWi:=word_to_int(TeW);(*把 TeW 转换为整型数*)
TeLi:=TeWi mod 256;(*求 TeWi 的低字节*)
TeHi:=TeWi/256;(*求 TeWi 的高字节*)
TeL:=int_to_word(TeLi);(*把 TeWi 的低字节的整型数转换为字节型 TeL*)
TeH:=int_to_word(TeHi) ;(*把 TeWi 的高字节的整型数转换为字节型 TeH*)
FCS:=TeH xor TeL;(*把 TeL 与 the 异或,其结果值即为所求 FCS*)
(*而这里所求的只是异或值,按 Host Link 协议,应把它转换为代表它的字符*)
(*但是,OMRON 公司的 ST 语言没有相应的转换函数,故只好在使用它时对它再做转换*)
```

提示：OMRON 公司新型 PLC 有 FCS 指令，可用以 FCS 计算。

在图 7-36 程序中，W0.00 ON 时，将调用 XFER 指令，把 D100 ~ D107 之间 7 个字的值，即上述命令码，依次传送给 D300 ~ D306。接着调用“ASC”指令，把在 D200 中存储 FCS 的值转换为 ASCII，并存入 D220 中。再调两次“MOVD”指令，实现把 FCS 两个字节加在命令码之后。之后，再调“逻辑或”及“MOV”，在命令字符串的最后，再加上字符“*”（十六进制值 2A）及回车符（十六进制值 0D0）。最终按 Host Link 协议，把编辑后的命令字符串存放在 D300 ~ D308（D308 仅使用其高字节）中。

有了合乎 Host Link 协议的通信命令字符串，调用命令发送指令，即可发送通信命令。如通信正确，则使对方 PLC 的 0100 通道的各个位置 1。当然，也可使 0100 字的各个位置不同的值。

提示：OMRON CJ 型 PLC 还支持 Modbus RTU 协议。可作为主站发送 Modbus RTU 协议命令，与对方作为 Modbus RTU 协议从站的 PLC 或只能装置通信。其细节见本书第 10.2 节。

提示：对被动方而言，协议通信的命令并不是 PLC 要执行的指令，而是一种能为 PLC 识别的操作要求。PLC 接收通信命令后，将按要求，与对方交换数据，或进行相应的操作。

7.2.5　PLC 与 PLC 网络协议通信编程

网络协议通信指令如表 7-8 所示。

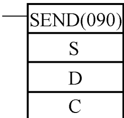
表 7-8　网络协议通信指令

网络发送	SEND	Explicit 读出指令	EGATR
网络接收	RECV	Explicit 写入指令	ESATR
指令发送	CMND	ExplicitCPU 单元数据读出指令	ECHRD
通用 Explicit 信息发送指令	EXPLT	Explicit ExplicitCPU 单元数据写入指令	ECHWR

以下仅对 SEND、RECV 及 CMND 指令做简要介绍，其它指令可参阅有关说明书。

1. SEND 指令

用于向网络节点发送数据。其梯形图格式为：



这里，S 为源字首地址，指明从本 PLC 哪个内存区读取数据；D 为目标字首地址，指明所读取的数据发送给哪个 PLC 的哪个内存区；C 为控制字首地址，指明要发送多少数据等信息，含义见表 7-9。

表 7-9　控制字 C 到 C+4 含义

字	00～07 位	08～15 位
C	字数:0001～允许最大值(4 位十六进制数字)	
C+1	目标网络地址	08～11 位:串行接口号 12～15 位:总为 0
C+2	目标单元地址	目标节点地址
C+3	重复次数:00～0F(0～15)	08～11 位:通信接口号 12～15 位:响应设置
C+4	响应监视时间:0001～FFFF(0.1～6553.5s) (默认设置 0000,设定监视时间为 2s)	

如图 7-39 所示，执行 SEND 指令，则把从本地节点的 S 地址开始的 C 指明的字数，传给目标节点的 D 地址开始的目标存储区中。

被传送的 PLC 不必编程。本指令也可微分执行。

图 7-40 所示为数据传送路径简图。

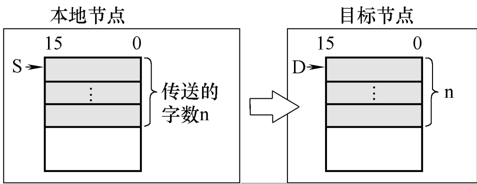


图 7-39　网络数据传送

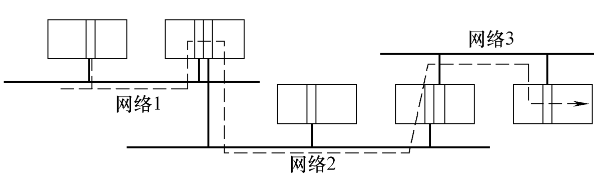


图 7-40　数据传送路径

如果目标节点号被设为 FF，数据将向指定网络的所有节点广播。这就是广播传送。

如果需要响应（C+3 的 12～15 位设置为 0），但在响应监视时间内未收到响应，数据最多可传输 15 次（在 C+3 的 0～3 位中设置重试次数）。

图 7-41 所示为发送数据梯形图程序。当“输入条件”和 A20200（对某机型端口 00 的通信端口允许标志）为 ON 时，D00100～D00109 的 10 个字被传输到本地网的节点 3D00000～D00009 的 10 个字中。如果在 10s 之内未收到响应，数据将传输 3 次。

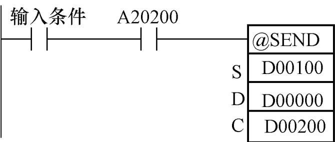


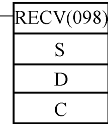
图 7-41　发送数据梯形图程序

这里 C 中各字的取值及含义如下：

C	D00200	0	0	0	A	要传送的字数:10个字
C+1	D00201	0	1	0	0	传输到网络0和串行通信板的端口1
C+2	D00202	0	0	1	0	节点号0 ,单元地址10
C+3	D00203	0	0	0	0	请求应答,端口号0,无重试
C+4	D00204	0	0	0	0	应答监视时间:2 s(0000:缺省值)

2. RECV 指令

用于从网络节点读取数据。其梯形图格式为：



这里，S 为源字首地址，指明从哪个内存区接收数据；D 为目标字首地址，指明所接收的数据存放在哪个内存区；C 为控制字首地址，指明要接收多少数，从哪个节点接收等信息，见表 7-9。

如图 7-42 所示，执行 RECV 指令时，请求把从源节点的字 S 开始的 C 中指定数目的字，传输到本地节点，并写入以 D 开始的数据区中。

图 7-43 所示为数据传送路径简图。

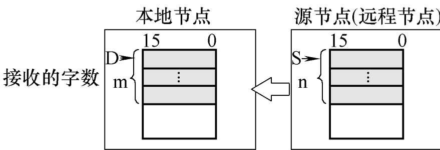


图 7-42　网络数据读取

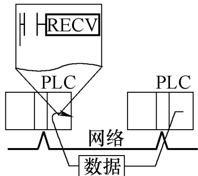
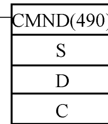


图 7-43　数据传送路径

RECV（098）要求有响应，因为响应包含要接收的数据。如果在 C + 4 中设置的应答监视时间内没有收到应答，数据传输请求重复达 15 次（重试次数在 C + 3 的 0 ~ 3 位设置）。

3. CMND 指令

用以向网络节点发送命令。其梯形图格式为：



这里，S 为源字首地址，指明从哪个内存区接收数据；D 为目标字首地址，指明所接收的数据存放在哪个内存区；C 为控制字首地址，指明要接收多少数，从哪个节点接收等信息，见表 7-10。

执行本指令，可向网络上的节点发送通信命令。用这些命令，可改变相关 PLC 的工作状态，如可改变指定的 PLC 处于监控工作模式等。

其通信过程如图 7-44 所示。

表 7-10 控制字 C 到 C+5 含义

字	00 ~ 07 位	08 ~ 15 位
C	命令数据的字节:0002 ~ 允许最大值(4 位十六进制)	
C+1	应答数据的字节:0000 ~ 允许最大值(4 位十六进制)	
C+2	目标网络地址:00 ~ 07	08 ~ 11 位:串行接口号 12 ~ 15 位:总是 0
C+3	目标单元地址:00 ~ FE	目标节点号: 00 ~ 允许最大值
C+4	重试次数:00 ~ 0F(0 ~ 15)	08 ~ 11 位:端口号 12 ~ 15 位:应答设置
C+5	应答监视时间:0001 ~ FFFF(0.1 ~ 6553.5s) (默认设置 0000,监视时间 2s)	

从图 7-44 可知，执行 CMND 指令，将通过 PLC 的 CPU 总线或网络，发送以字 S 为起始地址的指定字节数的 FINS 命令，到指定的设备。应答数据存储到以 D 开始的存储区中。

CMND 发送的是 OMRON FINS 协议的命令代码。如代码为 0102，那么执行 CMND 指令，如同执行 SEND；代码为 0101，执行 CMND 指令，如同执行 RECV。

例 1 图 7-45 所示的程序就是一个发送 FINS 命令到另一个 CPU 单元的例子。当 000000 和 A20207（某型机端口 07 的通信端口允许标志）为 ON 时，CMND 将 FINS 命令 0101（内存区读）传输到节点号 3，应答存储到 D00200 ~ D00211 中。

该命令从 D00010 ~ D00019 中读取 10 个字。应答包含有 2B 的命令代码（0101）、2B 完成代码，然后是 10 字的数据，总共 12 字或 24B。10s 内未接收到应答，数据将最多可重复传输 3 次。

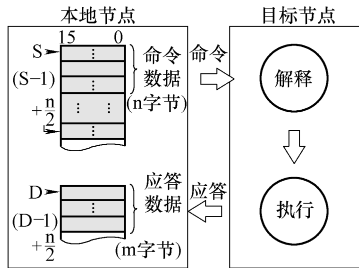


图 7-44 通信过程

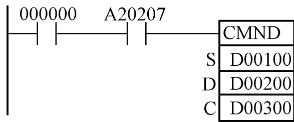


图 7-45 CMND 通信示例程序一

这里的 S 及 C 中各字的取值及含义如下：

字	15 ~ 8	7 ~ 0	含义
S:D00100	0 1	0 1	命令代码：0101 hex(内存区读)
S+1:D00101	8 2	0 0	数据区= 82 hex,地址= 000A00
S+2:D00102	0 A	0 0	
S+3:D00103	0 0	0 A	要读的字数= 0A hex(10十进制)

字	15 ~ 8	7 ~ 0	含义
C:D00300	0 0	0 8	命令数据的字节:0008(8十进制)
C+1:D00301	0 0	1 8	应答数据的字节:0018(24)
C+2:D00302	0 0	0 0	传输到本地网和设备本身
C+3:D00303	0 3	0 0	节点号3,单元地址00(CPU单元)
C+4:D00304	0 7	0 3	请求应答,端口号7,重试3次
C+5:D00305	0 0	6 4	应答监视时间 :0064十六进制(10s)

例 2　图 7-46 所示的程序显示了一个发送 FINS 命令到本地 CPU 单元的例子。当 CI/O 000000 和 A20207（某型机端口 07 的通信端口允许标志）为 ON，并且 A34313 为 OFF 时，CMND（490）将 FINS 命令 2215（创建/删除目录）传输到本地 CPU 单元。应答存储到 D00100～D00101 中。这里，FINS 命令将在 OMRON 目录下创建一个叫 CS/CJ 的目录。命令代码（2B）和结束代码（2B）将被返回，并作为应答存储。

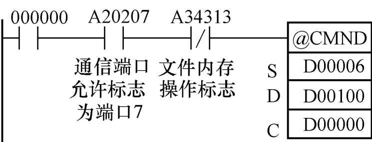


图 7-46　CMND 通信示例程序二

这里的 S 及 C 中各字的取值及含义如下：

	15～8	7～0	
S: D00006	2 2	1 5	命令代码:2215十六进制(创建/删除目录)
S+1: D00007	8 0	0 0	区号:8000十六进制(内存卡)
S+2: D00008	0 0	0 0	参数:0000十六进制(创建目录)
S+3: D00009	4 3	5 3	子目录名:CS1□□□□□.□□□ (□=空间)
S+4: D00010	3 1	2 0	
S+5: D00011	2 0	2 0	
S+6: D00012	2 0	2 0	
S+7: D00013	2 E	2 0	
S+8: D00014	2 0	2 0	
S+9: D00015	0 0	0 6	目录名长度:0006(6个字母)
S+10: D00016	5 C	4 F	绝对目录路径:/OMRON
S+11: D00017	4 D	5 2	
S+12: D00018	4 F	4 E	

	15～8	7～0	
C: D00000	0 0	1 A	命令数据的字节: 001A(26十进制)
C+1: D00001	0 0	0 4	应答数据的字节:0004(4)
C+2: D00002	0 0	0 0	目标网络地址:00十六进制(本地网)
C+3: D00003	0 0	0 0	目标单元地址:00十六进制目标节点号:00十六进制(本地节点的CPU单元)
C+4: D00004	0 7	0 0	请求应答,端口号7,重试0次
C+5: D00005	0 0	0 0	应答监视时间 :0000十六进制(6553.5s)

利用网络通信指令进行通信，过程较复杂，若被传的对方也正处于通信状态，则这个传送将不执行，故通信指令执行中，要求设定重试及其次数，还有不少成功或失败的标志。另外，被通信的对方也可设定或用指令（程序），予以禁止或保护，以保护自己的数据及自身安全。

7.3　PLC 与计算机通信编程（一）

关键词：数据读写、状态读写、主动通信、被动通信、协议通信、FINS、FINS C 模式、可视化编程软件、通信控件、OPC、电子邮件

PLC 与计算机通信，可通过串行接口，也可通过 PLC 网络或互联网。在习惯上，称计算机为上位机，而称 PLC 为下位机。

PLC 与计算机通信，如果为计算机方发起的则称为被动通信；而如果为 PLC 方发起的，则称为主动通信。大多数 PLC 与计算机通信为被动通信。

若为 PLC 主动通信，计算机与 PLC 双方都要运行程序，双方都要编程。但 PLC 方的程序基本上与 PLC 间通信程序相差不多。计算机方主要编写数据接收及必要的应答程序。

若为 PLC 被动通信，PLC 方无需编写通信程序，只需少量数据处理程序，编程重点在

计算机方。通信内容主要是数据及状态读写。数据读写或是计算机向 PLC 写数据，或是计算机从 PLC 读数据。状态写或是读 PLC 状态。写 PLC 状态，实际是计算机对 PLC 本身的控制，可使 PLC 停机（处于编程状态）或开机（监控或运行状态）。所以，对此类通信程序要慎重使用。

计算机通信程序类型很多，本节主要讨论可视化编程语言的编程，使用组态软件的编程将在 7.4 节讨论。

7.3.1　串行接口平台通信编程

利用串行接口平台进行 PLC 与计算机通信多是协议通信。以 OMRON PLC 为例，主要是用 Host Link（上位链接）协议，CS、CJ、及 CP 机还可用 FINS 协议。其实，不管什么协议，计算机程序只是通信命令的差异，而程序的结构、算法都是一样的。本节主要讨论 Host Link 协议编程。

1. HostLink 协议要点

OMRON HostLink 协议适用于所有它的 PLC 在串行接口平台上与计算机通信。通信由计算机向 PLC 发送命令，PLC 应答，PLC 为被动通信。这也是 PLC 与计算机之间用得最多的通信。在一次发送或应答中所含字符的集合称为帧。规定一个帧最多可含 131 个字符（节）。

1）帧格式。图 7-47 所示为计算机命令帧格式，其中每个方格为一个字符。

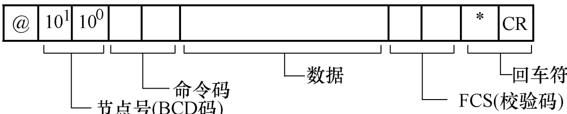


图 7-47　计算机命令帧格式

这里，@ 为命令开始字符。节点号用 BCD 码，2 位数，可在 00 ~ 31 之间选取，但必须与通信对方设定的节点号一致；命令码使用 2 个英文大写字母，代码解释见后；数据与命令码有关，数据的地址部分用 BCD 码，数据的数值部分用十六进制码，用到英文字符要大写；＊及 CR（回车符），为结束字符，是命令帧的结束标志。FCS 为异或校验，对 FCS 之前命令帧每个字符的 ASCII，按位依次异或，所得的结果值再换成 ASCII。此值不足 2 位数，高位要补 0，如用到英文字符也要大写。如下面一帧信息：@ 10（单元号）RH（命令）0031 0001（数据）58（FCS）＊CR（结束符）。这里的 FCS 为 58。这 58 是这么计算出来的：

```
  @(0100 0000,ASCII)
XOR(异或)
  1(0011 0001,ASCII)
XOR(异或)
  0(0011 0000,ASCII)
XOR(异或)
  R(0101 0010,ASCII)
  …
  0(0011 0000,ASCII)
XOR(异或)
  1(0011 0001,ASCII)
0101 1000(异或结果值为 58,十六进制数)
```

再转换为 ASCII 为：0011 0101（5）00111000（8）。

图 7-48 所示为 PLC 响应帧格式。它与命令帧基本相同，所差的只是在数据部分。这里增加了 2 个字符的返回码，如果命令正确，执行返回码为 00，不然将根据命令执行情况返回不同代码，而返回数据则不一定必要。写命令，就没有返回数据；读命令，才有返回数据。

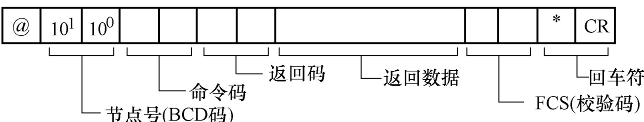


图 7-48 PLC 响应帧格式

提示：PLC 的节点号可使用 CX-Programmer 编程软件设定。出厂时，PLC 默认节点号为00。

2）多帧通信。如果通信交换的字符数超过 131 个字符，可以进行拆分而用多次通信，使每次都少于 131 个字符。也可使用多帧通信，如读、写程序，无法拆分。只能用多帧通信。多帧通信分为多帧响应及多帧命令，如图 7-49、图 7-50 所示。

如读很多数据，或读 PLC 程序，响应就是多帧响应。从图 7-49 可知，它的首响应帧没有结束“*”字符。计算机收到这个响应帧后，发应答符（回车符）；PLC 收到回车应答符后，再发后续数据，即中间帧，这样帧仅仅是数据、FCS 及回车符。计算机收到这个响应帧后，再发应答符（回车符）；PLC 收到回车应答符后，再发后续数据，即中间帧，但如果已是最后数据，那么按结束帧发送。结束响应帧与中间响应帧不同的是有“*”字符。计算机收到这样的帧就可做别的处理。响应大小由 PLC 自动生成，前面的帧为 128 个字符，结束帧为余下的不足 128 个字符。

如写很多数据，命令就是多帧命令。如图 7-50 所示就是多帧命令、应答及其结束响应的过程。与图 7-49 不同的是，它的命令帧大小由人工任意确定，只要不超过 128 个字符即可。

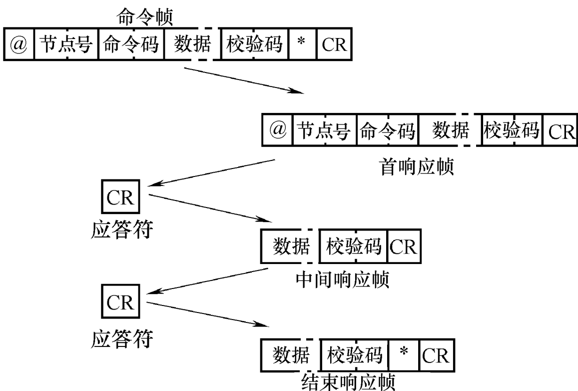


图 7-49 多帧响应

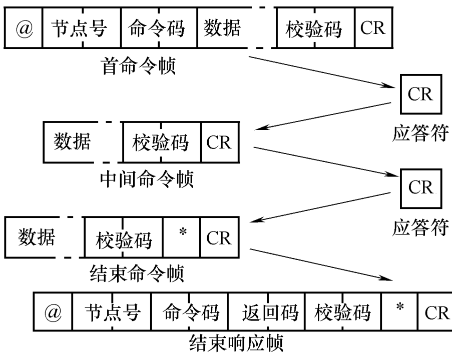


图 7-50 多帧命令

3）返回码。表 7-11 所示为主要的返回码。用好它，可便于通信程序的调试。

4）命令码。命令分为 3 个等级。等级 1 能读写 PLC 数据，若 PLC 处于监控及编程状态，还可向 PLC 写数据。等级 2 可向 PLC 传送程序，并可读写 I/O 表。等级 3 可进行 I/O 登记及 I/O 分布情况的读入。

表 7-11　返回码及其含义

返　回　码	含　　义	返　回　码	含　　义
00	命令正确执行	15	数据错误
01	RUN 模式无法执行	16	命令不支持
02	监控模式无法执行	18	帧长度错误
03	程序写保护	19	命令不可执行
04	地址超出	20	不能建 I/O 表
08	编程模式无法执行	21	CPU 单元错
13	FCS 错误	23	用户内存保护
14	命令格式错误		

一般用等级 1、2，但使用编程软件要用到等级 3。这 3 个命令等级可在上位链接单元作相应设定予以确定，如不用上位链接单元通信，一般为 3 级，而且随机型不同，可接收的通信命令也不尽相同。高档及新机型可接收的命令较多。

具体的通信命令可分为以下几类：

（a）测试类：命令码为 TS，用于对通信可行性进行测试。其格式为：

@ × × TS #####……FCS * CR

这里 × ×为 PLC 地址，可以是 00 ~ 31，由对 PLC 进行设定确定；#####为任意数字或字符；FCS 为纵向校验码，两个字符；＊为字符＊；CR 为回车。

计算机送这个命令给 PLC 后，若 PLC 返回的是同样的代码，则说明通信成功，可行；否则为通信不成功，不能正常进行通信。

这类命令常用于对通信硬件进行测试。

（b）数据读写类：命令码为 RX，或 WX，这里，X 为数据区符号，如 DM 区为 D，保持继电器为 H，辅助继电器为 J，计数器、定时器（现值）为 C，其设定值为#（或 $、或%）等。PLC 有多少内部器件就有多少相应的符号。

如为读命令，命令码后的数据先是指定读数据区的首地址，占 4 个字符；其后为要读的数据有多少字，也占 4 个字符，如读 1 个字符，为 0001。

如为写命令，命令码后的数据是写数据区的首地址，占 4 个字符；其后为依次向该数据区要写的内容，每个字占 4 个字符，要写多少字（通道），就有多少“4 个字符”。

接着为校验码，即 FCS，是两个字；最后为＊及回车符。

例 1　@00 RH 0000 0002 ＊CR

例 2　@00 WH 0000 FFFF FFFF ＊CR

这里，例 1 为要从 PLC 的 RH00 开始的 RH 区读 2 个字的内容。例 2 为要向 PLC 的 WH 区 WH00 开始的通道，依次写入 FFFF，FFFF 两个字的内容。

PLC 收到这两条命令后，如正确地执行了，其响应将分别为：

@00 RH 00 XXXX XXXX FCS ＊CR（对例 1）

这里的 XXXX XXXX 为 RH00 及 RH01 通道的数据。FCS 为校验码，占两个字符。

@00 WH 00 FCS ＊CR（对例 2）

这里，WH 后的 00 表示 WH 命令已正确执行。

应指出的是，数据写命令只能在监控及编程状态下才可能执行。

（c）PLC 状态读写：OMRON PLC 有 3 种状态，即编程、监控及运行。这 3 种状态可由 PLC 读写，其命令码分别为：MS 及 SC。

如果为状态读，其响应数据有两个字，各有其含义。而其第一个字的 08 及 09 位分别代表 PLC 的几种工作状态。如 08、09 位为

00 编程

10 运行

11 监控

如果为写状态，其数据仅一个字节，其 1、0 位的取值与要写的状态对应如下：

1 位 0 位

00 编程

10 监控

11 运行

（d）强迫置位与复位：其命令码分别为 KS、KR。

KS（强迫置位）的格式为：

@ × ×〖单元号〗KS〖置位命令〗× × × ×〖数据区〗× × × ×〖通道号〗× ×〖位号〗× ×〖置位值〗FCS * CR〖结束符〗

KR(强迫复位)的格式为：

@ × ×〖单元号〗KR〖复位命令〗× × × ×〖数据区〗× × × ×〖通道号〗× ×〖位号〗× × FCS * CR〖结束符〗

这里的数据区是指出强迫置位复位的内部器件名称，如 IR 区，为 CIO 加空格；LR 区为 LR 加两个空格；TIM 为 TIM 加空格；TIMH 为 TIMH 等。其对应关系见表 7-12。

表 7-12　数据对应关系

<table>
<tr><th rowspan="2">数据区</th><th colspan="4">操　作　数</th><th rowspan="2">通道号</th><th rowspan="2">位</th></tr>
<tr><th>OP1</th><th>OP2</th><th>OP3</th><th>OP4</th></tr>
<tr><td>IR or SR</td><td>C</td><td>I</td><td>O</td><td>空格</td><td>0000 ~ 0511</td><td rowspan="4">00 ~ 15</td></tr>
<tr><td>LR</td><td>LR</td><td>IR 或 SR</td><td>空格</td><td>空格</td><td>0000 ~ 0063</td></tr>
<tr><td>HR</td><td>HR</td><td>IR 或 SR</td><td>空格</td><td>空格</td><td>0000 ~ 0099</td></tr>
<tr><td>AR</td><td>AR</td><td>IR 或 SR</td><td>空格</td><td>空格</td><td>0000 ~ 0027</td></tr>
<tr><td>定时器</td><td>T</td><td>I</td><td>M</td><td>空格</td><td rowspan="5">0000 ~ 0511</td><td rowspan="6">00</td></tr>
<tr><td>高速定时</td><td>T</td><td>I</td><td>M</td><td>H</td></tr>
<tr><td>计数器</td><td>C</td><td>N</td><td>T</td><td>空格</td></tr>
<tr><td>可逆计数</td><td>C</td><td>N</td><td>T</td><td>R</td></tr>
<tr><td>累加计数</td><td>C</td><td>N</td><td>T</td><td>I</td></tr>
<tr><td>传送标记</td><td>T</td><td>N</td><td>空格</td><td>空格</td><td>0000 ~ 1023</td></tr>
</table>

强迫置位、复位是对该单元的直接赋值，不受程序影响，而数据写，则要受程序影响。强迫置位、复位只能在监控或编程方式下才能被执行。除了单点置位、复位，还有多点置

位、复位，其命令码分别为 FK、FR。另外，还有强迫置位、复位取消，其命令码为 KC，无操作数。

（e）程序读写：其命令码分别为 RP（读）、WP（写）。

RP 的格式为：

@ ××〖单元号〗RP〖命令码〗FCS〖校验码〗*CR〖结束符〗

其响应为：

@ ××〖单元号〗RP〖命令码〗××〖响应码〗××……〖程序机器码〗FCS〖校验码〗*CR〖结束符〗

WP 的格式为：

@ ××〖单元号〗WP〖命令码〗××……〖程序机器码〗FCS〖校验码〗*CR〖结束符〗

其响应为：

@ ××〖单元号〗WP〖命令码〗××〖响应码〗FCS〖校验码〗*CR〖结束符〗

PLC 的程序什么时候都可以读，但写只能在编程状态下进行。

（f）I/O 表读写：其命令码分别为 RI（读）、WI（写）。

RI 的格式为：

@ ××〖单元号〗MI〖命令码〗××〖 〗FCS *CR〖结束符〗

其响应为：

@ ××〖单元号〗MI〖命令码〗××〖响应码〗××……〖数据〗××〖FCS〗*CR〖结束符〗

WI 的格式为：

@ ××〖单元号〗MI〖命令码〗××……〖数据〗××〖FCS〗*CR〖结束符〗

其响应为：

@ ××〖单元号〗MI〖命令码〗××〖响应码〗××……〖数据〗××〖FCS〗*CR〖结束符〗

这两个命令用于 I/O 表登记。写只能在编程状态下才能进行。

（g）QQMR（登记）及 QQIR（读）：

QQMR 的格式为：

@ ××〖单元号〗QQMR〖命令码〗××××〖数据区符号〗×××××〖通道号〗××〖位号〗,〖分割符〗……××〖FCS〗*CR〖结束符〗

这里的分割符（,）之后还可有另一组数据，且一组之后还可另有一组。若之后没有新组，则不再插分割符，直接继之以 FCS 及 * CR。

其响应码为：

@ ××〖单元号〗QQMR〖命令码〗××〖响应码〗××〖FCS〗*CR〖结束符〗

QQIR 的格式为：

@ ××〖单元号〗QQIR〖命令码〗××〖FCS〗*CR〖结束符〗

其响应码为：

@ ××〖单元号〗QQIR〖命令码〗××……〖有关数据〗××〖FCS〗*CR〖结束符〗

这两条命令要配合使用。登记命令执行后登记的内容将一直保持，直到 PLC 掉电或再登记入新的内容，故登记之后，只要发简单的 QQIR 命令，即可成批地按登记的要求读 PLC 中不同器件的数据，非常方便。但这两条命令属于第三级命令。Host Link 单元需设成能执行三级命令时，它才能被执行。

（h）其它命令：还有其它一些通信命令，而且随着技术发展和新机型的出现，还将有新的命令推出，这一点一定要引起使用者注意。

其它命令中常用的有通信取消及通信初始化。它们的命令码分别为 XE（取消）、＊＊（初始化）。

XZ 命令的格式为：

@××〖单元号〗XZ〖命令码〗XX〖FCS〗*CR〖结束符〗

＊＊的格式为：

@××〖单元号〗**〖命令码〗XX〖FCS〗*CR〖结束符〗

这两条命令无响应码，常用于通信失败时进行再起动。

2. 计算机编程

VB、VC、Delphi 或 C＋＋Builder 等，都是可视化的编程软件，都可用以编写计算机通信程序。可视化编程可用通信控件编程，也可用 Windows 的通信 API 函数编程。

（1）VB6.0 编程。VB 编程使用通信控件比较方便。如用串行接口平台通信，有微软公司开发的串行接口通信控件（MSComm）；如用 Control Link 网通信，有 OMRON 公司提供的通信控件。安装好 VB、Control Link 网络管理软件后，这些控件会自动加入到 Windows 的动态库中。

在使用这些控件之前，要先把它从 VB 控件库中调入到 VB 的工具箱中。

1）MSComm 控件及其应用

a）CommPort（通信口）属性。指定用计算机的哪个串行接口进行通信。若用多个串行接口与不同的对象通信，则可多用几个控件。一个控件管理一个通信口。

b）Settings（参数设定）属性：指定串行接口的波特率、奇偶校验及停止位等特性。此设定应与 PLC 的设定一致。

c）Port Open（打开）属性：设置其为真，打开；否则关闭。进行通信之前，必须先打开。

d）InputMode（读取数据类型）属性：确定 Input 属性如何取回数据。设为 comInputModeText（默认），数据将以字符或字符串格式读出。设为 comInputModeBinary，数据将以二进制数用数组格式读出。如果数据只用 ANSI 字符集，则用 comInputModeText；否则用 comInputModeBinary。

e）Input（读取）属性：把输入缓冲区的内容读到指定的字符变量中。读后缓冲区的内容清为零。缓冲区的大小可设，最大可达 4k 字。

如果接收数据类型设为字符格式，则要先声明一个字符串变量，直接接收数据之后就可以显示它了。如：

```
Dim a As String                 '声明字符串变量
 a = MSComm1. Input             '接收数据
 Text1. Text = a                '在文本控件上显示字符串
```

如果接收数据类型设为二进制数，读取为数值是不可以视的；要先声明一个字节数组，读取后再转换为字符串才可以显示。如：

```
Dim b( ) As Byte                '声明字节数组
 l = MSComm1. InBufferCount     '确定接收数据的字节数
```

```
    b = MSComm1. Input                         '接收数据
    If l > 0 Then
       For i = 0 To l - 1                      '根据接收数据字节数把不可视的数字转换可视字符
         aa = hex(b(i))                        '把一个字节的十六进制数转换为两个字节的字符
         If Len(aa) = 1 Then aa = "0" + aa     '确保每个字节是两个字符
         a = a + aa                            '把单个字符组织成字符串
       Next i
       Text1. Text = a                         '在文本控件上显示字符串
  Else
       Text1. Text = ""                        '未接收到数据时文本控件上显示空
  End If
```

f）Output（输出）属性：把输出的命令送输出缓冲区，并且逐字向指定通信对象发送，直到缓冲区空。

发送的数据可以是可视的字符串，也可以是不可视的二进制数。

如果是可视的字符串，要先声明一个字符串变量，并进行赋值。然后发送这个字符串就可以了，即：

```
  Dim a As String                              '声明字符串变量
      a = "ABCD"                               '字符串变量赋值
      MSComm1. Output = a                      '发送字符串
```

如果发送的数据类型设为二进制数，因为它是数值，是不可视的，要先声明一个字节数组，再把可视的数值转换为不可视的二进制数，并赋值给所声明的字节数组，然后发送这个字节数组，即

```
  Dim b( ) As Byte                             '声明字节数组
  Dim a As String                              '声明字符串变量
  a = "ABCD"                                   '十六进制数用的字符串变量赋值
  l = Len(a$)                                  '计算字符串包含的字节数
    For k = 0 To l - 2 Step 2                  '把可视的字符串转换为字节数组,两个十六进制字符存于一
                                                个字节中
    b(k\2) = Val("&H" + Mid(a, k + 1, 2))
      Next k
      MSComm1. Output = b                      '发送字节数组
```

提示：发送、接收使用二进制方式，虽然它的数据不可视，不太方便，但通信效率高。同样的帧长，而收、发的数据可增大一倍。更何况有的协议，如 Modbus RTU 还必须使用二进制格式通信。

g）RThreshold（接收字符数）属性：设定值为整型数。如果设置为 0（默认值），接收缓冲区接收到字符不产生 OnComm 事件。如果设置为 1，每收到一个字符会使 MSComm 控件产生 OnComm 事件。其余类推。

h）OnComm（ ）事件属性：这个控件只有这一个事件。只要 Comm Event（也是一种属性）的值发生变化，On Comm（ ）事件就会发生。

Comm Event 取值与通信口的状况有关，有 18 个可能的取值。其中 10 个与通信口的出错（如缓冲区满、通信时间超出……）有关；而 7 个与通信的进程（如输入或输出缓冲区

内容改变……）有关。如什么事件也不发生，则其值为零。

2）通信编程使用的方法：

（a）程序访问方法。它用程序定时或依需要访问事件属性或访问缓冲区计数值（InBufflrCount 或 OutBuff Count）的变化，依其变化情况作相应的通信处理。以下以读 PLC DM0000 及 0001 的值为例说明这个编程。

本例使用的窗口为 Form1，其上加载的控件有：MSComm1（通信控件，未特别指明其特性用默认值）、Text1（文本控件，用以显示所得到数据）及 Command1（按钮控件，用以激发 Command1_ Click 过程，实现通信），如图 7-51a 所示。

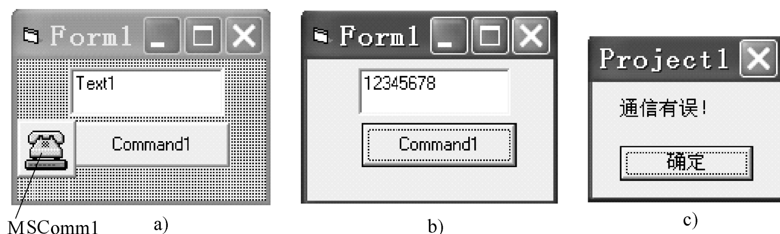


图 7-51　程序访问方法显示窗口及控件

程序代码如下（“'”之后为注解）：

```
Private Sub Command1_Click( )
    MSComm1. CommPort = 1                      '使用计算机串行接口 1
    MSComm1. Settings = "9600,e,7,2"           '通信速率为 9600bit/s、偶校验、2 位停止位
    MSComm1. PortOpen = True                   '口打开
    o$ = "@00RD00000002"                       'FCS 之前命令帧
    oo$ = o$ + fcs(o$) + "*"                   'FCS 之前命令帧 + FCS 校验码 + "*"
    MSComm1. Output = oo$ + Chr(13)            '再加回车后,发送
    tt = Timer                                 '记录当前时间
    Do                                         '等待
     Dummy = DoEvents( )                       '释放控制权,允许 CPU 处理其它作业
     ll = MSComm1. InBufferCount               '记录缓冲区已收到字符数
    Loop Until ll > = 11 Or Timer > tt + 0. 2  '已收到 19 个字符或等待 0. 2s 跳出循环
    Instring = MSComm1. Input                  '读缓冲区,接收数据
    If Mid$(Instring, 6, 2) = "00" Then        '如果通信无误
     Text1. Text = Mid$(Instring, 8, 8)        '显示数据
    Else                                       '否则通信有误
     MsgBox"通信有误!"                          '提示通信有误!
    End If
    MSComm1. PortOpen = False                  '关闭串行接口
End Sub
```

而 FCS 函数代码为：

```
Public Function fcs(a$)
  b% = 0                                       '初始化为 0
  l% = Len(a$)                                 '计算命令 FCS 之前帧字符串长度
```

```
    For I% =1 To l%                          'fcs check
    b% =b% Xor Asc(Mid(a$, I% , 1))          '按位依次异或
    Next I%
    ff$=Hex$(b%)                             '转换为字符
    If Len(ff$) =1 Then
    ff$="0" +ff$                             '如果 FCS 字符串仅 1 位,高位补"0"
    End If
      fcs =ff$                               '函数输出
  End Function
```

运行本程序，点击“Command1”按钮，如果通信正常，Text1 将显示所读数据，如 7-51b 所示为 12345678。如果通信不正常，将弹出图 7-51c 所示的对话框，显示“通信有误!”

（b）事件驱动方法。所用说明实例及使用控件同上。程序代码如下（“’”之后为注解）：

```
  Private Sub Command1_Click()
      MSComm1.CommPort =1                    '使用计算机串行接口 1
      MSComm1.Settings ="9600,e,7,2"         '通信速率 9600bit/s、偶校验、2 位停止位
      MSComm1.PortOpen =True                 '口打开
      MSComm1.RThreshold =19                 '当输入缓冲区收到 19 个字符激发接收事件
      o$ ="@00RD00000002"                    'FCS 之前命令帧
      oo$ =o$ +fcs(o$) +"*"                  'FCS 之前命令帧 +FCS 校验码 +"*"
      MSComm1.Output =oo$ +Chr(13)           '再加回车后,发送
  End Sub
    Private Sub MSComm1_OnComm()
      If MSComm1.CommEvent =2 Then           '当确认为接收事件执行以下代码
          Instring =MSComm1.Input            '接收字符
          Text1.Text =Mid$(Instring, 8, 8)   '显示有效字符
          MSComm1.PortOpen =False            '关闭口
          MSComm1.RThreshold =0              '串口状态复原
      End If
  End Sub
```

执行本程序效果与上述访问方法相同。只是如果通信不正常，将没有提示；必要时可增加 Timer 控件予以监视；超出监视时间还没有回应，也可提示出错。

（2）Delphi6.0 编程。Delphi 是 Inprise 公司推出的基于对象 Pascal 语言的可视化集成开发工具。也可使用微软公司提供的 MSComm 控件，只是在使用前要以“ActiveX Control”的身份加载到它的集成编程平台上，然后把这控件拖放到程序所用的窗口（FORM）中。

图 7-52a 所示为 Delphi6.0 通信程序实例的运行窗口。实例除了使用 MSComm 控件，还有其它标准控件，如图 7-52b 所示。

从图 7-52 可知，其上有：6 个“Button”（按钮），分别为，open（打开通信口）、关闭通信口（close）、fcs（FCS 校验）、send（发送）、receive（接收）及 exit（程序退出）的按钮。还有用以键入通信命令的“Edit1”（编辑文本框，用以输入通信命令）、“Edit2”（编辑文本框，用以显示或键入命令原码 + 校验码 + 结束码）及一个显示 PLC 应答数据的

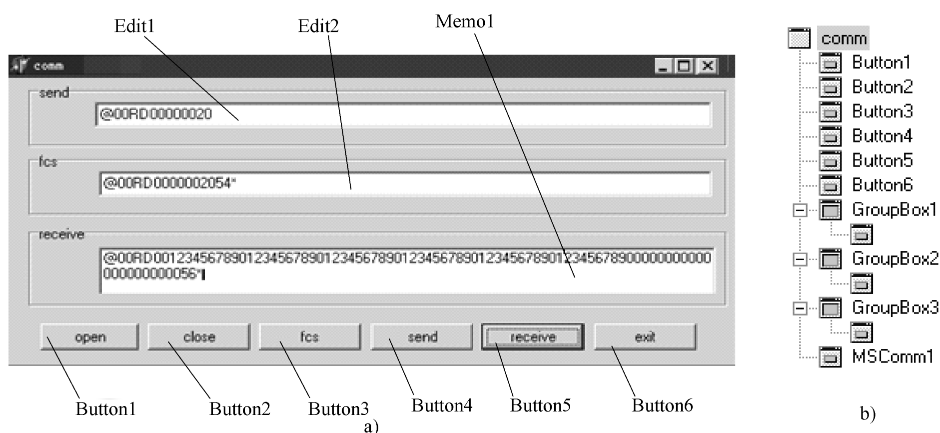


图 7-52　Delphi6.0 通信程序实例
a）运行窗口　b）所用控件

“Memo1”（文本框）。微软 MSComm 控件运行时是看不见的，编辑时对它的属性设定与 PLC 的串行接口特性一致。

以下为本程序代码。显示 Delphi 的“开头”代码是由系统生成的。用浅色字符表示。其余为各个按钮点击后执行的程序代码，由用户编写，并含有简要注解。有 6 个程序段分别与 6 个按钮对应。

```
unit delphi_mscomm;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, OleCtrls, MSCommLib_TLB;
type
    Tcomm = class(TForm)
    MSComm1:TMSComm;
    GroupBox1:TGroupBox;
    Edit1:TEdit;
    GroupBox2:TGroupBox;
    Edit2:TEdit;
    GroupBox3:TGroupBox;
    Button1:TButton;
    Button2:TButton;
    Button3:TButton;
    Button4:TButton;
    Button5:TButton;
    Button6:TButton;
    Memo1:TMemo;
    procedure Button1Click(Sender:TObject);
    procedure Button6Click(Sender:TObject);
//  procedure Button7Click(Sender:TObject);
```

```
      procedure Button2Click(Sender:TObject);
      procedure Button4Click(Sender:TObject);
      procedure Button5Click(Sender:TObject);
      procedure Button3Click(Sender:TObject);
    private
      { Private declarations }
    public
      { Public declarations }
    end;
var
    comm:Tcomm;
implementation
{$R *.dfm}
procedure Tcomm.Button1Click(Sender:TObject);          '与 open 按钮对应
        begin                                           '如果串口未打开,则打开串口
            if not mscomm1.portOpen then
              mscomm1.PortOpen:=true;
            end;
procedure Tcomm.Button2Click(Sender:TObject);          '与 close 按钮对应
      begin                                             '如果串行接口打开,则关闭串行接口
      if  mscomm1.portOpen then
            mscomm1.PortOpen:=false;
      end;
procedure Tcomm.Button3Click(Sender:TObject);          '与 fcs 按钮对应
                                                        '用以计算 FCS 校验码
      var                                               '定义变量
      ii,i,k,N:integer;
      bb,B:integer;
      C:string;
      ComStr,A:string;
      ch:string[255];
      begin
       I:=1;
       b:=0;
      ComStr:=edit1.Text;                               '读入通信命令
      K:=length(ComStr);                                '计算命令字符串长度
      ch:=Comstr;
      for I:=1 to k do                                  '按位依次异或
      begin
      //  b:=b xor VarToInt(ch[I]);//VarToStr(copy(ComStr,I,1)));
          for ii:=42 to 97 do
              begin
              if AnsiCompareStr(Chr(ii),ch[i])=0 then
```

```
                    bb：= ii；
                  end；
            b：= b xor bb；
        end；
        c：= intTohex(B,2)；  //返回两位
        // edit3. text：= varToStr(c)；
          if length(c) = 1 then
            c：='0' + c；
            c：= c + '*'；
            edit2. text：= edit1. Text + c；               '计算结果输出到 Edit2 文本框中
        end；
    procedure Tcomm. Button4Click(Sender:TObject)；     '与 send 按钮对应
        begin                                           '命令输出
          Mscomm1. Output：= edit2. text + chr(13)；
        end；
    procedure Tcomm. Button5Click(Sender:TObject)；     '与 receive 按钮对应
        var
          recstr:string；                               '定义变量
        begin
          recstr：= mscomm1. input；                    '接收数据
          Memo1. Text：= recstr；                       '接收数据显示到 Memo1 中
        end；
    procedure Tcomm. Button6Click(Sender:TObject)；     '与 exit 按钮对应
        begin
        close；                                         '用以打开串口
        end；
```

运行本程序后，只要在“命令原码”处键入“@ 00RD00000020”通信命令（读 DM0000 开始的 10 个字的数据），然后单击“fcs”按钮，再依次单击“open”按钮，“send”按钮、“receive”按钮，如通信正常，将看到如图 7-53 所示的情况。输入其它命令将有不同显示。要退出系统，先单击“close”按钮，后再单击“exit”按钮即可。

（3）VB. Net 编程。VB. Net 包装在微软新版本的 Visual Studio 2005 开发工具中。使 VB 提升为也可面向对象及多线程编程。在它的 . Net Framework 2. 0 类库中，包含 SerialPort 类（用于串行接口通信），类似 VB6. 0 的 MSComm 控件，但功能却有很大提升，可更方便地用于串行接口通信程序编程。

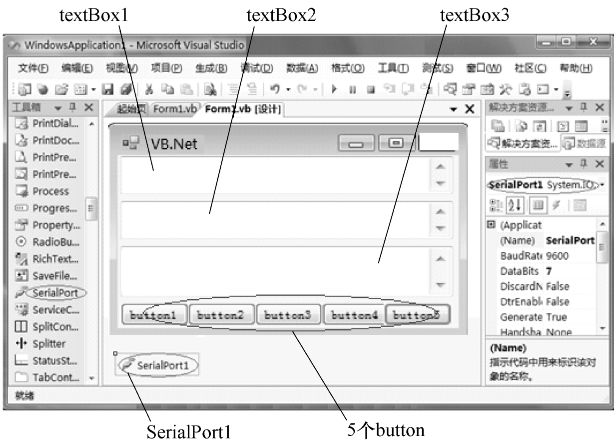


图 7-53　VB. Net 开发环境及 Form1 窗口

图 7-53 所示为 VB. Net 的开发环境及建立的一个工程“VB. Net”窗口。

从图 7-53 可知，在 VB. Net 窗口上，加载了 5 个 button 按钮控件及 3 个 textBox 文本控件。前者用于激发执行与通信相关的程序，后者用以键入与显示通信数据。此外，还有串口通信控件 SerialPort1。

这 5 个 button1 ~ button5 分别用于打开串行接口、关闭串行接口、起动 FCS 校验码计算、发送通信命令及接收通信数据。textBox1 ~ textBox3 分别用于键入通信命令、存储 FCS 运算结果、显示接收数据。

程序作为 Form1 类包装在一起，其有关代码及必要注释如下。

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If SerialPort1.IsOpen = False Then SerialPort1.Open()
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        If SerialPort1.IsOpen = True Then SerialPort1.Close()
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        Dim bb, LL, II As Integer
        Dim ff As String
        bb = 0                                          '初始化为
        LL = Len(TextBox1.Text)                         '计算命令 FCS 之前帧字符串长度
        For II = 1 To LL                                'fcs check
            bb = bb Xor Asc(Mid(TextBox1.Text, II, 1))  '按位依次异或
        Next II
        ff = hex$(bb)                                   '转换为字符
        If Len(ff) = 1 Then
            ff$ = "0" + ff$                             '如果 FCS 字符串仅为位,高位补"0"
        End If
        TextBox2.Text = TextBox1.Text + ff + " * "      '命令输出
    End Sub
        Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
        SerialPort1.Write(TextBox2.Text + Chr(13))      '发送命令
    End Sub
    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click
        TextBox3.Text = SerialPort1.ReadTo(Chr(13))     '接收数据,直到收到回车符(Chr(13))
    End Sub
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    '初始化串行接口,也可在控件属性表中设定,此程序在串行接口加载时运行
        SerialPort1. PortName = "COM1"
        SerialPort1. Parity = IO. Ports. Parity. Even
        SerialPort1. StopBits = 2
        SerialPort1. BaudRate = 9600
    End Sub
End Class
```

图 7-54 所示为本程序运行画面。这时，如在 textBox 中键入“@00RD00000020”通信命令（读 DM0000 开始的 10 个字的数据），然后单击“button3”按钮，再逐次单击“open”按钮、“button4”按钮、再单击“button5”按钮。如通信正常，将看到如图 7-54 所示的情况。输入其它命令将有不同显示。要结束通信，可单击“button2”按钮来关闭通信口。要退出系统，可单击窗口右上角的“✕”键。

（4）C#编程。C#编程语言是微软在微软新版本的 Visual Studio 2005 中，新开发的编程语言。具有 VB 界面好、开发快及 VC 程序效率高、运行速度快的双重优点。也是与 SUN 公司 JAVA“PK”的产物。它也自带有通信控件 SerialPort，可很方便地用以编程写通信程序。

图 7-55 所示为 C#的开发环境及建立的一个工程“C#”窗口。

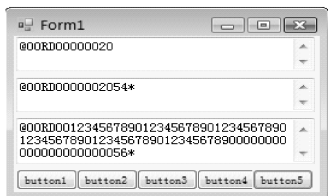


图 7-54　通信程序运行画面

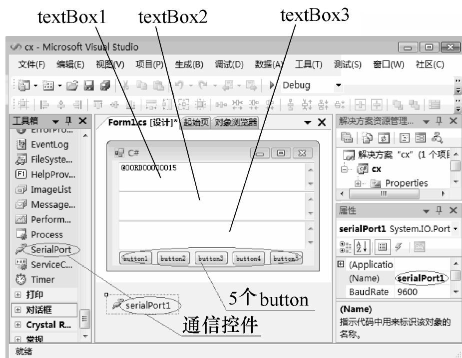


图 7-55　C#开发环境及 Form1 窗口

从图 7-55 知，在 C#窗口上，与上述 VB. Net 窗口一样，也加载了 5 个 button 按钮控件及 3 个 textBox 文本控件。前者用于激发执行与通信相关的程序，后者用以键入与显示通信数据。此外，还有串行接口通信控件 SerialPort1。

这 5 个 button1 ~ button5 分别用于打开串行接口、关闭串行接口、起动 FCS 校验码计算、发送通信命令及接收通信数据。textBox1 ~ textBox3 分别用于键入通信命令、存储 FCS 运算结果和显示接收数据。

程序关代码及必要注释如下。

```
using System;
using System. Collections. Generic;
using System. ComponentModel;
using System. Data;
using System. Drawing;
using System. Text;
using System. Windows. Forms;
```

```
namespace WindowsApplication6
{
    public partial class Form1:Form
    {
        public Form1()
        {
            InitializeComponent();            '用于初始化及加载窗口上所有控件
        }
        //以上各代码是系统生成的。以下为用户键入的
        private void Form1_Load(object sender, EventArgs e)
        {
            serialPort1.PortName = "COM1";'使用串行接口1
            serialPort1.Parity = System.IO.Ports.Parity.Even;'偶校验
            serialPort1.StopBits = System.IO.Ports.StopBits.Two; '2位停止位
            serialPort1.BaudRate = 9600; '波特率为9600bit/s
        }
private void button1_Click(object sender, EventArgs e)
        {
            if(serialPort1.IsOpen = = false)
                serialPort1.Open();'如果串行接口未打开,则打开
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if(serialPort1.IsOpen = = true)
                serialPort1.Close();'如果串行接口打开,则关闭
        }

        private void button4_Click(object sender, EventArgs e)
        {
                serialPort1.Write(textBox2.Text + "\r"); '发送通信命令
        }

        private void button5_Click(object sender, EventArgs e)

        {
                textBox3 .Text = serialPort1.ReadTo("\r");'接收数据,直到回车符
        }
        private void button3_Click(object sender, EventArgs e)
        {'计算校验码FCS
            int ll,bb = 0;
            string aa,ff;
            aa = textBox1.Text;'读入通信命令
```

```
            ll = aa. Length; ' 计算命令字符串长度
            char[ ] chars = aa. ToCharArray( ) ;' 命令转换为字符数组

            for( int i = 0;i  <  ll;i + + )
                bb = bb^chars[ i] ;  ' 按位依次异或
            string s = " " ;
            s = String. Format( " {0:X} " ,bb) ; //转换为十六进制数的字符串
            if( s. Length    = = 1 )
              s = '0' + s;  ' 如果字符串不到 2 位,高位加“0“
            textBox2. Text = textBox1. Text + s + " * " ;' 输出到 textBox2
        }
    }
}
```

本程序运行与图 7-54 所示的 VB. Net 的运行画面相同。如在 textBox 中键入“@00RD00000020”通信命令（读 DM0000 开始的 10 个字的数据），然后单击“button3”按钮，再依次单击“button1”按钮、“button4”按钮、“button5”按钮，要结束通信，也可单击“button2”按钮。要退出系统，可单击窗口右上角的“”键。

（5）VC 编程。VC 通信程序编程使用 API（Application Program Interface，应用程序接口）函数比较方便。

1）API 函数是微软公司在设计 Windows 操作系统时加进去的。使 Windows 除了协调应用程序的执行、分配内存、管理系统资等之外，还成了一个应用服务中心，为应用编程提供了很大方便。串行接口通信用 API 函数主要有：

CreateFile,打开串行接口；

GetCommState,读取串行接口通信参数；

SetCommState,设置串行接口通信参数；

BuilderCommDCB,用字符串中的值来填充设备控制块；

GetCommTimeouts,读取通信超时参数；

SetCommTimeouts,设置通信超时参数；

SetCommMask,设置被监控事件；

WaitCommEvent,等待单个被监控事件发生；

WaitForMultipleObjects,等待多个被监测对象的结果；

WriteFile,发送数据,即写串行接口；

ReadFile,接收数据,即读串行接口；

GetOverlappedResult,返回最后重叠(异步)操作结果；

PurgeComm,清空串行接口缓冲区,退出所有相关操作；

ClearCommError,更新串行接口状态结构体,并清除所有串行接口硬件错误；

CloseHandle,关闭串行接口。

2）串行接口通信程序要点。串行接口通信有两种操作方式：同步操作方式和重叠操作方式（又称为异步操作方式）。同步操作时，调用的 API 函数会阻塞直到操作完成以后才能返回（在多线程方式中，虽然不会阻塞主线程，但是仍然会阻塞监听线程）；而重叠操作方式中 API 函数会立即返回，在后台进行操作，避免线程的阻塞。无论哪种操作方式，

一般都通过 4 个步骤来完成，即打开串行接口、配置串行接口、读写串行接口及关闭串行接口。

（a）打开串行接口。

用 API 函数 CreateFile 来打开或创建。该函数的原型为：

```
HANDLE CreateFile(
    LPCTSTR lpFileName,
    DWORD dwDesiredAccess,
    DWORD dwShareMode,
    LPSECURITY_ATTRIBUTES lpSecurityAttributes,
    DWORD dwCreationDistribution,
    DWORD dwFlagsAndAttributes,
    HANDLE hTemplateFile
);
```

其中：

lpFileName：将要打开的串行接口逻辑名，如“COM1”；

dwDesiredAccess：指定串行接口访问的类型，可以是读、写或读写共三种方式；

dwShareMode：指定共享属性，由于串行接口不能共享，该参数必须置为 0；

lpSecurityAttributes：引用安全性属性结构，默认值为 NULL；

dwCreationDistribution：创建标志，对串行接口操作该参数必须置为 OPEN_EXISTING；

dwFlagsAndAttributes：属性描述，用于指定该串行接口是否进行异步操作 FILE_FLAG_OVERLAPPED，表示使用异步的 I/O；该值为 0，表示同步 I/O 操作；

hTemplateFile：对串行接口而言该参数必须置为 NULL；

（b）配置串行接口。

在打开串行接口，获得通信设备句柄后，需要对串行接口进行配置。为此，要用到 DCB（Device Control Block，设备控制块）结构。DCB 结构包含了串行接口通信的各项参数设置，诸如波特率、数据位数、奇偶校验和停止位数等信息。下面仅介绍该结构几个常用的参数：

```
Typedef struct _DCB{
…
DWORD BaudRate;                    '波特率,指定通信的传输速率
DWORD fParity;                     '指定奇偶校验使能
…
BYTE ByteSize;                     '每字节的位数
BYTE Parity;                       '奇偶校验方法
BYTE StopBits;                     '停止位的位数
…
} DCB;
```

用 CreateFile 打开串行接口后，可以调用 GetCommState 函数来读取串行接口通信初始参数。

即：

```
BOOL GetCommState(
HANDLE hFile,                      '通信端口的句柄
LPDCB lpDCB                        '指向 DCB 结构的指针
);
```

如果需要修改通信参数，可先修改 DCB 结构，然后再调用 SetCommState 函数实现所作的更改。即：

```
BOOL SetCommState(
HANDLE   hFile,                    '通信端口的句柄
LPDCB   lpDCB                      '指向 DCB 结构的指针
);
```

除了上述设置，一般还要设置 I/O 缓冲区大小和超时值。Windows 用 I/O 缓冲区来暂存串行接口输入和输出的数据，调用 SetupComm 函数可以设置串行接口的输入和输出缓冲区的大小。即：

```
BOOL SetupComm(
HANDLE hFile,                      '通信端口的句柄
DWORD dwInQueue,                   '输入缓冲区的大小(字节数)
DWORD dwOutQueue                   '输出缓冲区的大小(字节数)
);
```

在读写串行接口时，需要考虑超时问题。超时的作用是指在指定的时间内没有读取或发送指定数量的字符，读写仍然会结束。

要查询当前的超时设置应调用 GetCommTimeouts 函数，该函数会填充一个 CommTimeouts 结构。调用 SetCommTimeouts 可以用某一个 CommTimeouts 结构的内容来设置超时。经验证明 CommTimeouts 结构中各个参数的设置将会影响到通信效率。在保证正确通信的前提下，各个参数的值越小，通信速度越快。

（c）读写串行接口。

可以使用 ReadFile 和 WriteFile 读写串行接口，下面是这两个函数：

```
BOOL ReadFile(
    HANDLE hFile,                      '串行接口的句柄
    LPVOID lpBuffer,                   '读取数据存储的地址指针
                                       '即读取的数据将存储在该指针指向的内存区
    DWORD nNumberOfBytesToRead,        '需要读取串行接口数据接收缓冲区的字节数
    LPDWORD lpNumberOfBytesRead,       '指向一个 DWORD 指针,返回
                                       '实际读取的字节数
    LPOVERLAPPED lpOverlapped,         '重叠执行时,该参数指向一个 OVERLAPPED
                                       '结构,同步执行时,该参数为 NULL
    );
BOOL WriteFile(
    HANDLE hFile,                      '串行接口的句柄
    LPCVOID lpBuffer,                  '写入的数据存储的地址指针
    DWORD nNumberOfBytesToWrite,       '需要写入串行接口数据发送缓冲区的字节数
    LPDWORD lpNumberOfBytesWritten,    '指向指向一个 DWORD 指针,返回实际写
                                       '入的字节数
    LPOVERLAPPED lpOverlapped          '重叠执行时,该参数指向一个 OVERLAPPED 结构
                                       '同步执行时,该参数为 NULL
    );
```

在使用 ReadFile 和 WriteFile 读写串行接口时，既可以同步执行，也可以重叠执行。同步执行时，函数直到操作完成后或设置的超时时间到才返回，这意味着当前线程被阻塞，从而导致效率下降。重叠执行时，即使操作还未完成，这两个函数也会立即返回，费时的 I/O 操作可以在后台进行。

如果操作成功，这两个函数都返回 TRUE。需要注意的是当 ReadFile 和 WriteFile 返回 FALSE 时，不一定就是操作失败，线程应该调用 GetLastError 函数分析返回的结果。例如，在重叠操作时如果操作还未完成函数就返回，那么函数就返回 FALSE，而且 GetLastError 函数返回 ERROR_ IO_ PENDING，这说明重叠操作还未完成。

（d）关闭串行接口。

利用 API 函数关闭串行接口非常简单，只需使用 CreateFile 函数返回的句柄作为参数调用 CloseHandle 即可：

```
BOOL CloseHandle(
HANDLE hObject;                              '串行接口的句柄
);
```

3）VC 串行接口通信编程实例。VC 用 Api 函数编写通信程序上述 4 要点的代码，当然还得有个装载容器。这容器最简单是用对话框。图 7-56 所示为对话框。其上有：5 个“按钮”，3 个“文本框”。按钮分别为：Open Port（打开通信口）、Close Port（关闭通信口）、FCS（FCS 校验）、Send（发送）及 Receive（接收）。文本框分别为：“命令原码”，用以输入通信命令；“命令原码 + 校验码及结束码”，用以显示 FCS 计算后的命令码；“接收字符”，用以显示接收到的字符。

图 7-56　通信对话框

程序中 5 个按钮函数的代码见下：

（a）串行接口建立及打开：

```
void CHyCommDlg::OnOpenPort()//对应“Open Port”按钮,打开串行接口
{
  m_hComm = CreateFile("COM1",
                      GENERIC_READ | GENERIC_WRITE,
                      0,
                      NULL,
                      OPEN_EXISTING,
                    FILE_ATTRIBUTE_NORMAL|FILE_FLAG_OVERLAPPED,
                      0);
  if (m_hComm = =INVALID_HANDLE_VALUE)
  {
      AfxMessageBox("COM1 Port can't be opened!");
  }
  else
```

```
    {
        // Success for opening COM port,then we must set state of
        // COM device.
        DCB dcb;
        FillMemory( &dcb, sizeof( dcb) , 0) ;
        if ( ! GetCommState( m_hComm, &dcb) )       // get current DCB
        {
            // Error in GetCommState
            AfxMessageBox( "Error in GetCommState" ) ;
        }
        else
        {
            // Update DCB rate.
            dcb. BaudRate = CBR_9600  ;
            dcb. ByteSize = 7 ;
            dcb. Parity    = EVENPARITY;
            dcb. StopBits = TWOSTOPBITS;
            // Set new state.
            if ( ! SetCommState( m_hComm, &dcb) )
            {
                // Error in SetCommState. Possibly a problem with the communications
                // port handle or a problem with the DCB structure itself.
                AfxMessageBox( "Error in SetCommState" ) ;
            }
            else
            {
                // Success in SetCommstate
                CreateReceiveThread( ) ;
            }
        }
    }
}
```

（b）FCS 计算：

```
void CHyCommDlg::OnFcs( )                    //对应“FCS”按钮,准备发送命令存于 m_strSrc
{
    UpdateData( TRUE) ;
    CString str = m_strSrc;
    FCS( str) ;
    m_strFcs = str;
    UpdateData( FALSE) ;
} void FCS( CString & str)    //加上校验码( FCS)后的报文存于 m_strFcs
{
    CString o,oo,ff,fcs,o1;
```

```
    int b,l,i;
    char o2;
    o = str;
    b = 0;
    l = o. GetLength( );

    for( i = 0;i < l;i ++ )
    {
          o2 = o. GetAt( i );
          b = b^o2;
    }
    ff. Format( "%X",b);                    //要用大写字母
    if( ff. GetLength( ) == 1)
          ff = '0' + ff;
    fcs = ff + " * ";
    oo = o + fcs;
    str = oo  + " \r";
}
```

（c） 发送数据：

```
void CHyCommDlg::OnSend( )                //对应“Sent”按钮,发送读数据命令,即发 m_strFcs
{
    DWORD dwActualWrite;
    OVERLAPPED o = {0};
    BOOL fRes;
    DWORD dwRes;
    if(! WriteFile( m_hComm,( LPCTSTR) m_strFcs,m_strFcs. GetLength( ),&dwActualWrite,&o) )
          {
               if (GetLastError( )! = ERROR_IO_PENDING) {
                    // WriteFile failed, but isn't delayed. Report error and abort.
                    fRes = FALSE;
               }
               else
                    // Write is pending.
                    dwRes = WaitForSingleObject( o. hEvent, INFINITE);
               switch( dwRes)
               {
                    // OVERLAPPED structure's event has been signaled.
               case WAIT_OBJECT_0:
                    if (! GetOverlappedResult( m_hComm, &o, &dwActualWrite,FALSE) )
                    {
                         fRes = FALSE;
                    }
                    else
```

```
                {
                    // Write operation completed successfully.
                    fRes = TRUE;
                }
                break;

            default:
                // An error has occurred in WaitForSingleObject.
                // This usually indicates a problem with the
                // OVERLAPPED structure's event handle.
                fRes = FALSE;
                break;
            }
            }
            else
            {
                // WriteFile completed immediately.
                fRes = TRUE;
        }
    }
```

(d) 接收数据:

```
void CHyCommDlg::OnReceive()                //对应"Receive"按钮,接收数据
{
CString s;
  int ll;
char odw;
DWORD dwActualRead;
OVERLAPPED o = {0};
      CString str = m_strSrc;
odw = str.GetAt(3);
if(odw == 'W')                              //如果是写命令,接收字符长度为 11
  ll = 11;
else
{
str = str.Mid(9,4);
  ll = atoi( str ) * 4 + 11;                //如读命令,从命令中读接收的字符长度
}
LPSTR p = s.GetBuffer(ll);
ReadFile(m_hComm,p,ll,&dwActualRead,&o);    //接收数据
s.ReleaseBuffer();                          //接收数据
m_strReceived = s;                          //显示接收数据
UpdateData(FALSE);                          //显示接收数据
}
```

（e）串口关闭：

```
void CHyCommDlg::OnClosePort()        //对应"Close"按钮,关闭串行接口
{
if(m_hComm)
{
        CloseHandle(m_hComm);
        m_hComm = NULL;
}
if(m_hRcvThread)
{
        CloseHandle(m_hRcvThread);
}
if(m_dwRcvThreadID)
{
        m_dwRcvThreadID = 0;
}
}
```

运行本程序，在“命令原码”处键入“@00RD00000010”通信命令（读 DM0000 开始的 10 个字的数据），然后单击“FCS 校验（FCS）”按钮，再单击“打开通信口（Open Port）”按钮，再单击“Send（发送）”按钮，再单击“Receive（接收）”按钮，再单击“Close Port（关闭通信口）”按钮，如通信正常，将看到如图 7-56 所示的情况。

VC 通信程序可处理成多线程的，即在前台处理其它工作的同时另建立一个线程，在后台处理通信（VB 只好用中断了），这既提高了工作速度，又可作到程序的其它任务处理与通信两不误。

7.3.2　网络平台通信编程

PLC 利用 PLC 网络平台与计算机通信也多是协议通信。OMRON PLC 主要是用 FINS（Factory Interface Network Service）协议，也称信息服务通信命令（Message Service Communications Commands），此外，还可运用数据链接通信。

1. FINS 协议要点

FINS 通信命令可用以不同的控制操作，如发送及接收数据、改变 PLC 的工作模式、强制置位与复位、文件操作等等。它是应用层的协议，所以可用于不同网络。只是如果用于以太网或串行接口平台，其命令及响应帧要增加相应的头部（headers）及尾部。

图 7-57a 所示为 FINS 命令帧格式。图 7-57b 所示为 FINS 响应帧格式。

这里，ICF（Information Control Field，信息控制域）一个字节，其取值见图 7-57c。RSV（Reserved，保留字节）常为 00；GCT（Gateway Count：Number of Bridges Passed Through，历经网络数）常为 02hex；DNA（Destination network address，目标网络地址）在 00（本地网络号）~7F（127）选定；DA1（Destination node address，目标节点地址）在 00（PLC 内部通信）~20hex 之间选定，如选 FFhex，为广播传送；DA2（Destination unit address，目标单元地址）在 00（CPU 单元）~FEhex 之间选定。SNA（Source network address，源网络地

址）在 00（本地网络）~7Fhex 之间选定。SA1（Source node address，源节点地址）在 00（PLC 内部通信）~20hex 之间选定。SA2（Source unit address，源单元地址）在 00（CPU 单元）及 10hex ~ 1Fhex（CPU 总线单元地址，为 10hex + 单元号）之间选定。SID（Service ID，服务 ID）用以指定生成的过程，在 00 ~ FFhex 之间选定。命令码占 2 个字节，不同取值有不同含义，见表 7-13。Text 为参数，有地址，有数据，多少字节取决于命令码。End 为返回码，占 2 个字节，反映命令执行的情况，如 00 为正常执行。

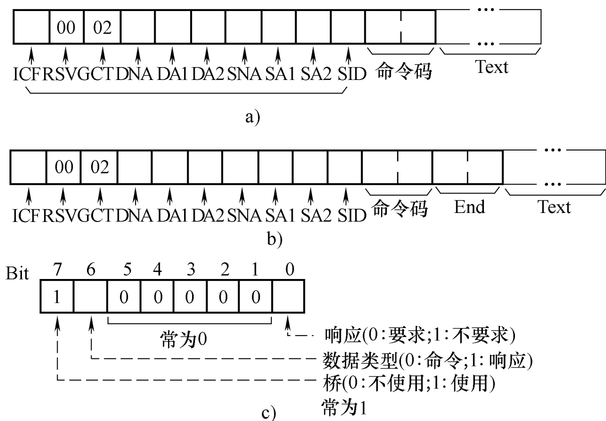


图 7-57　FINS 格式
a）FINS 命令帧格式　b）FINS 响应帧格式　c）ICF 格式

FINS 命令码很多。表 7-13 所示为 FINS 部分命令码及其含义。命令码高字节为 MR（Main，主），低字节为 SR（Sub，辅）。如 MR 01 为 I/O 内存区访问。而怎么访问由 SR 确定。从表知，它有 5 种访问方式：读、写、填充、多处读及传送。

表 7-13　FINS 部分命令码及其含义

类　型	命令码		名　称	功　能
	MR	SR		
I/O 内存区访问	01	01	读内存区	指定区按字连续读
	01	02	写内存区	指定区按字连续写
	01	03	填充内存区	指定区写相同值
	01	04	多内存区读	指定区非连续读
	01	05	内存区数据传送	复制一指定连续区数据到另一指定连续区
参数区访问	02	01	参数区读	指定区按字连续读
	02	02	参数区写	指定区按字连续写
	02	03	参数区填充	指定区写相同值
程序区访问	03	06	程序区读	读 UM(用户内存)区
	03	07	程序区写	写 UM(用户内存)区
	03	08	程序区清除	清除 UM(用户内存)区
调试	23	01	置位、复位强制	位置位、复位强制及取消
	23	02	强制置位、复位取消	取消所有强制状态

在命令帧中，参数用到的地址很多，且与 PLC 型号有关。表 7-14 所示为部分 CS、CJ 型 PLC 地址代号。

从表知，参数占 4 个字节。数据或操作类型一个字节，地址编号 3 个字节。头 2 个字节为字地址，取值为 0 到字可能的最大的地址，后一个字节为位地址，取值为 0 ~ F。所有地址值都是用十六进制数。

在参数中，数据值的指定或表示，位用 1 个字节，01hex 为 ON，00hex 为 OFF；字用十六进制数，按实际数表示。数据强制时，每个位作为一个元素。每一元素用一个字节表示。字节中 00 位表示指定为数据，01 位表示强制状态。字也可强制写。

返回码也是分有主辅 2 个部分。表 7-15 所示为部分返回码及含义。

表 7-14　部分 CS/CJ 型 PLC 地址代号

区　域		数据:类型	CS/CJ 型 PLC			长度(字节)
			内存区代号(hex)	内存区地址	内存地址	
CIO Area	CIO	Bit	30	CIO 000000 ~ CIO 614315	000000 ~ 17FF0F	1
Work Area	WR		31	W00000 ~ W51115	000000 ~ 01FF0F	
Holding Bit Area	HR		32	H00000 ~ H51115	000000 ~ 01FF0F	
CIO Area	CIO	Word	B0	CIO 0000 ~ CIO 6143	000000 ~ 17FF00	2
Work Area	WR		B1	W000 ~ W511	000000 ~ 01FFF00	
DM Area	DM	Bit	02	D000000 ~ D3276715	000000 ~ 7FFF0F	1
	DM	Word	82	D00000 ~ D32767	000000 ~ 7FFF00	2
CIO Area	CIO	Bit with forced status	70	CIO 000000 ~ CIO 614315	000000 ~ 17FF0F	1
Work Area	WR		71	W00000 ~ W51115	000000 ~ 01FF0F	
Holding Bit Area	HR		72	H00000 ~ H51115	000000 ~ 01FF0F	

表 7-15　部分返回码及含义

主码	辅码	检查点	可能原因	纠错建议
00:正常执行	00:正常执行	…	…	…
	01:服务取消	…	服务被取消	检查指定区第三节点的可能
		数据链接状态	服务被取消	检查链接状态
01:局部节点出错	01:局部节点不在网络	局部节点状态	局部未接入网络	连接该节点
	02:令牌超时	最大节点地址	令牌没有到达	设定节点地址小于在最大值
	03:重试故障	…	指定次数发送不成功	做通信测试,检查系统
	04:发送帧太多	允许发送帧数	超出最多事件帧数	检查网络每周期执行帧数,增加最多事件帧数
	05:节点地址超出	节点地址	节点地址设定出错	重设地址,确保地址惟一
	06:节点地址重复	节点地址	节点地址设定出错	重设地址,确保地址惟一

图 7-58 所示为一组命令帧与响应帧实例。图 7-58a 为命令，7-58b 为响应。

本命令含义是读 PLC DM000A 开始的 10 个字数据。返回码为 00，意即命令已正确执行，并返回所读 10 个字数据。

FINS 协议相比 Host Link 协议，功能要强的多，如可进行“位”操作，可进行 4 位以上地址字（如 DM10000）操作，可在运行模式可修改数据，帧长度可达 1000 个字符，可跨网络中继操作等。

对 CS、CJ、CP 型 PLC，FINS 命令还可在串行接口平台使用。但在上述格式的基础上，要增加头及尾，如图 7-59 所示。这也称为 FINS C 模式。

图 7-59a、7-59b 所示格式用于计算机串行接口与 PLC 串行接口通信命令及响应，图

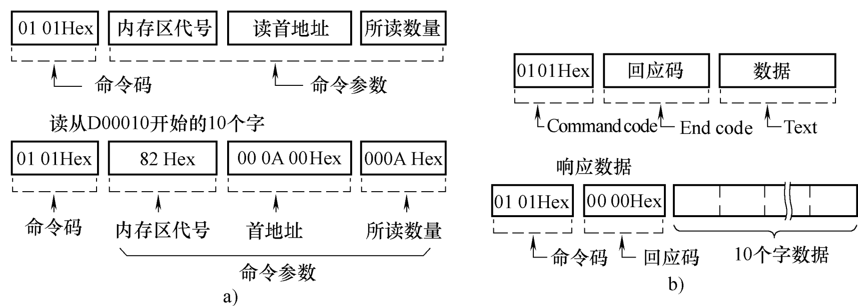


图 7-58　一组 FINS 命令帧与响应帧实例

a）命令　b）响应

7-59c、7-59d 所示格式用于计算机串行接口与连接网络上的 PLC 通信命令及响应。

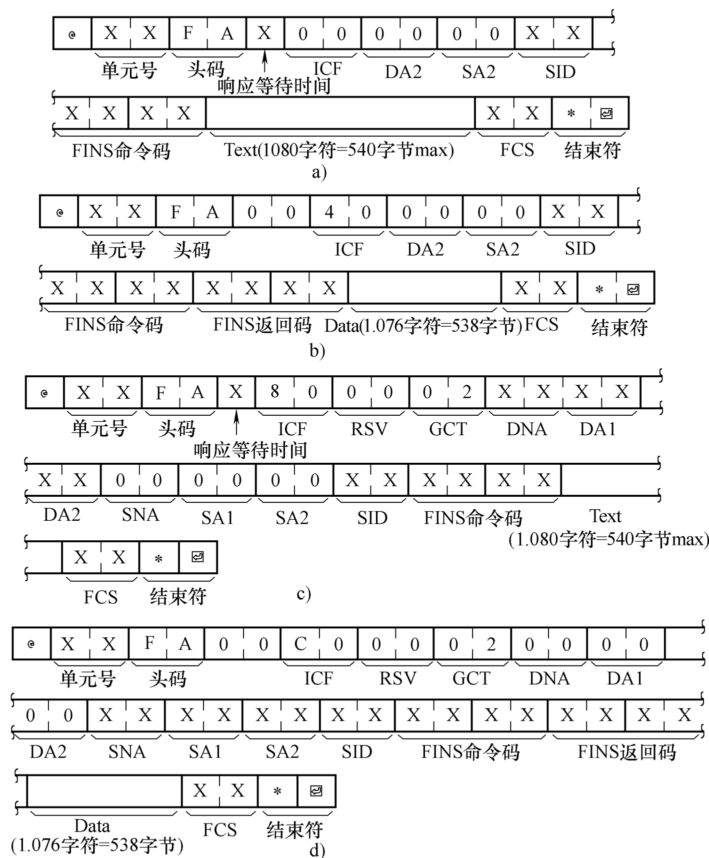


图 7-59　FINS C 模式计算机发命令 PLC 响应

a）计算机串行接口与 PLC 串行接口通信计算机命令　b）计算机串行接口与 PLC 串行接口通信 PLC 响应

c）计算机串行接口与连接网络上的 PLC 通信计算机命令　d）计算机串行接口与连接网络上的 PLC 通信 PLC 响应

这里，有关字节含义与上述介绍的 Host Link 及 FINS 的相同。FINS 协议命令用十六进制数，而 Host Link 协议用 ASCII，故 FINS 用作 Host Link 通信时，要把命令中的十六进制数

转换为 ASCII。如值“0”应为 30hex，值“A”应为 41hex，等等。所以，用它通信比直接用 FINS 通信，同样多通信字节，信息量要少一半。

图 7-60 所示为一通信实例。可用以说明网络地址、节点地址及单元地址。

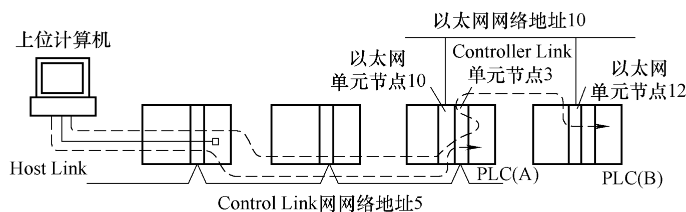


图 7-60　通信实例

如图，若从计算机发送命令到在网络 5 上节点 3 PLC（A）的 CPU 单元，那么它的（DNA）：05（30，35）、（DA1）：03（30，33）、（DA2）：00（30，30）。若从计算机发送命令到在网络 10 上节点 12 PLC（B）的 CPU 单元，那么它的（DNA）：0A（30，41）、（DA1）：0C（30，43）、（DA2）：00（30，30）。

FINS C 模式也可由 PLC 发命令，计算机响应。这时不仅可使用 CMND 指令，还可使用 SEND 及 RECV 指令。上述指令的数据还是按指令规则处理，但计算机收到的格式将与上述发送时类似。而且，计算机的回应也应与计算机发送时的格式相同。

以上只是 FINS 命令的简要介绍。还有很多细节，请参阅 OMRON 公司提供的《SYSMAC CS/CJ Series Communications Commands REFERENCE MANUAL》。

2. 计算机编程

FINS 协议在计算机方，可以用于串行接口，也可用于以太网（卡），还可用于 Control Link 网（卡），等等。前两者可用可视化软件编程，后者可用 OMRON 通信工具软件提供的控件编程。

（1）串行接口 FINS 协议通信编程。它与 Host Link 完全相同。所不同的只是命令及响应的内容有所差别。以前介绍的所有可视化通信程序均可使用。

如读取 CP1H 型 PLC DM20000 开始 2 个字的数据，可发送：@00FA0800002000000000FC000101 824E2000000203 * 回车，如果在 PLC 中，DM20000 的值为 1234hex，DM20001 的值为 5678hex。其响应将是：@00FA00C000020000FC0000000001010000 12345678 3B * 回车

如果向 CP1H 型 PLC DM32000、32001 2 个字写“12345678”数据，可发送：@00FA0800002000000000FC000102 827 D00000002 12345678 08 * 回车。如正确执行将响应：@00FA00C000020000FC000000000102 0000 30 *

如果要置位 W1.1，可发送@00FA000000000010231000101 0001 01 76 * +回车。如置位成功其响应为：@00FA00400000000102 0000 40 *

如读取 CIO 10.13 位数据，可发送：@00FA00000000001013 0000A0D 0001 70 * 回车。如该位 ON，则响应为：@00FA004000000001010000 01 42 *

如置位 CIO 10.13，可发送：@00FA000000000010230000A0D00010073 *，如正确写了，其响应为：

@00FA00400000000102 0000 40 *

如果强制 W100.00 置位，可发送：@00FA00000000023010001 0001 3100640077 * 回车。如强制成功其响应为：@00FA00400000002301000043 *

如果强制 W100.00 复位，可发送：@00FA00000000023010001 0000 3100640076 * 回车。如强制成功其响应为：@00FA00400000002301 0000 43 *

如果要清除所有强制，可发送：@ 00FA000000000230274 * 回车，如清除成功，则响应为：@00FA00400000002302 0000 40 *

提示：使用 FINS 协议可访问地址超过 10000 的 D 区。所以，对 CS、CJ 及 CP 型 PLC 的监控程序应尽量使用此协议。只是，它用的是十六进制码，功能又太强，要小心使用，否则易出错。

提示：FINS 协议读写多少数据都要用十六进制码指明。而 Host Link 协议，读多少数用 BCD 码指明，写多少数不用指明。

（2）以太网通信编程。要使用 TCP（传输控制协议）或 UDP（用户数据报协议）协议。前者须先建立连接，才可发送、接收数据，通信较为可靠。后者无需建立连接即可传送数据，比较简便，但不大可靠。与 OMRON PLC 以太网通信编程也使用这两个协议，而且还要与 FINS 相结合。

TCP、UDP 通信要用到的 Socket。Socket 是由美国伯克利大学开发的在 Unix 系统上的通信编程规范，用于计算机通信则代表一种点到点数据传输。通信双方由代表两点的“服务器”和“客户端”组成，基于 IP 协议进行按照 TCP 或 UDP 规范进行信息交换。

建立双方通信的过程即称为建立一个“Socket（套接字）”，建立后利用得到的“套接字”进行各种信息的交流。随着 Windows 系统的流行，开始有人在原来的基础上移植到 Windows 平台上。微软在早期编写了基于 Windows 特征的（消息驱动等）“套接字”编程 API，一般称其为“Winsock API”。此外，还有“Winsock”控件。

1）使用 Winsock API 编程

a）以太网 TCP 协议通信用 API 函数主要有：

socket：创建套接字；

listen：监听；

accept：请求连接；

connect：建立连接；

send：发送数据；

rec：接收数据；

closesocket：关闭套接字。

b）以太网 TCP 协议通信程序要点。

对服务器端其要点是：

用 socket 函数，创建套接字；

用 bind 函数，将套接字绑定到一个本地 IP 地址及端口上；

用 listen 函数，将套接字设为监听模式，准备接收客户请求；

等待客户请求到来，如果到来，用 accept 函数，接收请求，并返回一个新的对应于此次连接的套接字。

用 send 或 recv 函数，通过这个新的套接字，从客户端读取数据或向客户端发送命令；返回，等待另一个客户连接。

如果不再通信，关闭套接字。

对客户端其要点是：

用 socket 函数，创建套接字；

用 connect 函数，向服务器发出连接请求；

用 send 或 recv 函数，通过套接字，向服务器发送命令或从服务器读取数据；

如果不再通信，关闭套接字。

c）程序实例。在 OMRON 以太网操作手册（W421-E1-03）中，有一个 C 语言无连接以太网通信程序实例。在 UNIX 工作站上运行，以发送 FINS 命令（命令码为 0101），读取连接在以太网的 PLCD00100 开始的 150 字数据。设定的 PLC 以太网单元 IP 地址是 196.36.32.100，FINS 节点地址是 100，IP 地址转换设为自动，FINS UDP 口为缺省，9600。工作站 IP 地址为 196.36.32.50，节点地址是 50，UDP 口设为 0，动态分配。同时，如果设定在 2 秒内未接到回应，通信命令将重发。具体程序转引于此，供参考。

```
#include <errno. h>
#include <stdio. h>
#include <sys/types. h>
#include <sys/socket. h>
#include <netinet/in. h>
#include <signal. h>
#define FINS_UDP_PORT 9600
#define SERV_IP_ADDR "196.36.32.100" /* Ethernet Unit IPADDRESS */
#define MAX_MSG 2010
#define RESP_TIMEOUT 2
/*
 * FINS COMMUNICATIONS SAMPLE PROGRAM
 */
main(argc,argv)
int argc;
char *argv[];
{
int sockfd;
struct sockaddr_in , ws_addr,cv_addr;
char fins_cmnd[MAX_MSG],fins_resp[MAX_MSG];
int sendlen,recvlen,addrlen;
char sid =0;
```

```
extern recv_fail( ) ;

/ * GENERATE UDP SOCKET * /
if( ( sockfd = socket( AF_INET,SOCK_DGRAM,0) ) ≪0)
err_exit( " can't open datagram socket" ) ;

/ * ALLOCATE IPADDRESS AND PORT # TO SOCKET * /
bzero( ( char * ) &ws_addr,sizeof( ws_addr) ) ;
ws_addr. sin_family = AF_INET;
ws_addr. sin_addr. s_addr = htonl( INADDR_ANY) ;
ws_addr. sin_port = htons( FINS_UDP_PORT) ;
if( bind( sockfd, ( struct sockaddr * ) &ws_addr,sizeof( ws_addr) ) < 0)
err_exit( " can ' t bind local address" ) ;

/ *
 * GENERATE MEMORYAREA READ COMMAND
 * ( READ 150 WORDS FROM D00100. )
/ *
fins_cmnd[ 0 ] = 0x80; / * ICF * /
fins_cmnd[ 1 ] = 0x00; / * RSV * /
fins_cmnd[ 2 ] = 0x02; / * GCT * /
fins_cmnd[ 3 ] = 0x01; / * DNA * /
fins_cmnd[ 4 ] = 0x64; / * DA1 * / / * Ethernet Unit FINS NODE NUMBER * /
fins_cmnd[ 5 ] = 0x00; / * DA2 * /
fins_cmnd[ 6 ] = 0x01; / * SNA * /
fins_cmnd[ 7 ] = 0x32; / * SA1 * / / * WS FINS NODE NUMBER * /
fins_cmnd[ 8 ] = 0x00; / * SA2 * /
fins_cmnd[ 9 ] = + + sid; / * SID * /
fins_cmnd[ 10 ] = 0x01; / * MRC * /
fins_cmnd[ 11 ] = 0x01; / * SRC * /
fins_cmnd[ 12 ] = 0x82; / * VARIABLE TYPE: DM * /
fins_cmnd[ 13 ] = 0x00; / * READ STARTADDRESS: 100 * /
fins_cmnd[ 14 ] = 0x64;
fins_cmnd[ 15 ] = 0x00;
fins_cmnd[ 16 ] = 0x00; / * WORDS READ: 150 * /
fins_cmnd[ 17 ] = 0x96;


/ * SEND FINS COMMAND * /
bzero( ( char * ) &cv_addr,sizeof( cv_addr) ) ;
cv_addr. sin_family = AF_INET;
cv_addr. sin_addr. s_addr = inet_addr( SERV_IP_ADDR) ;
cv_addr. sin_port = htons( FINS_UDP_PORT) ;
```

```
singnal((SIGALRM,recv_fail);

CMND_SEND:
sendlen = 18;
if(sendto(sockfd,fins_cmnd,sendlen,0,&cv_addr,sizeof(cv_addr))
 = = sendlen){
alarm(RESP_TIMEOUT); /* START RESPONSE MONITOR TIMER */
printf("send length %d¥n",sendlen);
}
else{
err_exit("send error");
}

/* RECEIVE FINS RESPONSE */
if((recvlen = recvfrom(sockfd,fins_resp,MAX_MSG,0,&cv_addr,&addrlen))
 <0){
if(errno = = EINTR)
goto CMND_SEND; /* RE-SEND FINS COMMAND */
err_exit("receive error");
}
else{
alarm(0); /* STOP RESPONSE MONITOR TIMER */
printf("recv length %d¥n",recvlen);
if(recvlen < 14) /* ILLEGAL RESPONSE LENGTH CHECK */
err_exit("FINS length error");
if((fins_cmnd[3]! =fins_resp[6])||(fins_cmnd[4]! =fins_resp[7])
||(fins_cmnd[5]! =fins_resp[8])){ /
* DESTINATION ADDRESS CHECK */
err_exit("illegal source address error");
}
if(fins_cmnd[9]! =fins_resp[9]) /* SID CHECK */
err_exit("illegal SID error");
}


/* CLOSE SOCKET */
close(sockfd);
}
/*
 * ERROR PROCESSING FUNCTIONS
 */
err_exit(err_msg)
```

```
char  * err_msg;
{
printf("client: %s %x ¥n",err_msg,errno);
exit(1);
}

/ *
 * SIGNAL CAPTURE FUNCTIONS
 * /
recv_fail( )
{
printf("response timeout error  ¥n");
}
```

2）Winsock 控件编程。Winsock 控件为 VB 使用“套接字”编程提供了方便。利用 Winsock 控件可以与通信对方建立连接，并通过传输控制协议（TCP）进行数据交换。

a）Winsock 控件属性、方法及事件。其主要属性有：

BytesReceived：返回当前缓冲区中接收到的字节数量。

LocalHostName：返回本机名字符串。

LocalIP：返回以（xxx. xxx. xxx. xxx）格式表达的 IP 地址串。

LocalPort：本机使用的地址，可读写，设计时可用，Long 型。对于客户，如果不需要指定端口，则用端口 0 发送数据。在此情况下，控件将随机选择一个端口。在一个连接确定后，成为 TCP 的端口。对于服务器，指用于监听的端口。如设置为 0，则用随机数。在调用 Listen 方法后，该属性自动包含用到的端口。端口 0 总是用于在两计算机间建立动态连接。客户希望通过端口 0 获得一个随机端口以“回调”连接服务器。

Protocol：套接字类型，为 TCP 或 UDP 二者之一，默认为 TCP 类型。设置为 sckTCPProtocol 表示 TCP 协议 sckUDPProtocol 表示 UDP 协议。在此属性被重置之前需用 Close 方法关闭。

RemoteHost：发送或接收数据的主机，你可提供主机名如：" FTP：//ftp. microsoft. com"；或一 IP 地址串，例如" 100. 0. 1. 1"。

RemoteHostIP：远程主机的 IP 地址。对于客户程序，在连接确定后使用 Connect 方法，此属性包含远程主机的 IP 名串。对于服务器程序，在引入连接需求后（ConnectionRequest 事件)，此属性包含 IP 串。当使用 UDP 套接字，在 DataArrival 事件发生后，此属性为发送 UDP 数据的机器 IP 地址串。

RemotePort：连接套接字端口值。例如通常 HTTP 应用使用 80 端口，FTP 则使用 21，LEC G3 机的端口为 502。

State：控件状态，只读。可为以下值：0（sckClosed)，默认值，关闭；1（sckOpen)，打开；2（sckListening)，侦听；3（sckConnectionPending)，连接挂起；4（sckResolvingHost)，识别主机；5（sckHostResolved)，已识别主机；6（sckConnecting)，正在连接；7（sckConnected)，已连接；8（sckClosing)，同级人员正在关闭连接；9（sckError)，错误。

主要方法有：

Accept：仅用于 TCP 服务器应用。这个方法用于在引入一个连接时响应的事件，即 Con-

nectionRequest 事件。语法：object. AcceptrequestID 返回值：Void。响应事件时必须传递 RequestID 参数给此方法，以生成新的 Socket 实例用于实际的信息传输。

Bind：设定 LocalPort 及 LocalIP 用于 TCP 连接。当有多个协议适配器时使用。语法：object. BindLocalPort，LocalIP。LocalPort，此端口用于连接，LocalIP 生成连接的 IP 地址。如果已设定相关属性，可不必携带相关参数。在调用 Lisent 方法之前调用此方法。

Close：在客户或服务器方关闭 TCP 连接。语法：object. Close。参数：无。返回值：Void。

GetData：接收存于可变类型中的数据块。返回值：Void。语法：object. GetDatadata，[type,] [maxLen]。Data：接收数据的变量，如果空间不够，将设置为空。Type：可选，接收的类型，自行设置。MaxLen：可选参数。设定接收数组或字符串类型数据的尺寸。如果参数没有，将接收所有的数据。如果提供数组或字符串以外的数据类型，则忽略此参数。Type 可以设置为常用的数据类型。通常在 DataArrival 事件中使用该方法。此事件包含 totalBytes 参数。如果你设定的 maxlen 小于 totalBytes 参数，你将得到一个由 10040 表示的剩余字节将丢失的警告信息。

Listen：建立一个设置为监听模式的套接字。此方法仅用于 TCP 连接。语法：object. Listen 参数：无。返回值：Void。当调用 Listen 之后，引入一个连接时发生 ConnectionRequest 事件。当处理 ConnectionRequest 时，应用程序必须使用 Accpet 方法来响应。

PeekData：同 GetData 类似，但是 PeekData 不从输入队列中移去数据。此方法仅用于 TCP 连接。语法：object. PeekDatadata，[type,] [maxLen]。

SendData：向远地主机发送主机。返回值：Void。语法：object. SendDatadata。Data：发送的数据为二进制数，使用字节数组。当使用 Unicode 格式串时将在发送之前转换为ANSI 串。

Connect：客户机端可以用此方法请求与服务器连接。连接前应指定远程服务器 IP 地址及端口号。但也可以调用此方法是指定。

发生的事件有：

Close：发生于远程主机关闭连接。为了正确的关闭 TCP 连接应当使用 Close 方法。

Connect：当连接行动完成时，语法：object. Connect ( )。用此事件表明连接成功。

ConnectionRequest：发生于一个远端主机要求确定一个连接时。仅用于 TCP 服务器应用。RemoteHostIP 及 RemotePort 属性在此事件后存储了关于客户机的信息。语法：object_ConnectionRequest (requestIDAsLong)。requestID：引入的连接的请求标识。此参数传给 Accept 方法中的第二个控件实例。服务器可以决定是否认可该连接。如果引入的连接未被认可，客户将受到一个 Close 事件。使用 Accept 方法接受引入的连接。

DataArrival：当新数据抵达时发生。语法：object _ DataArrival (bytesTotalAsLong)。BytesTotal，Long 型。总计收到的数据量。此事件在你调用 GetData 方法之前将不会再发生。仅在有新数据抵达时激活。你可以在任何时刻使用 BytesReceived 属性检查多少数据有效。

Error：表明发生了错误。限于篇幅，错误码忽略。

SendComplete：当发送动作完成时发生。语法：object_SendComplete。参数：无。

SendProgress：当发送数据时产生本事件。语法：object_SendProgress (bytesSentAsLong，bytesRemainingAsLong)。BytesSent：本事件发生以来发送的数据量。BytesRemaining：缓冲区中等待发送的数据。

b）计算机与 PLC 以太网通信的计算机 VB 程序实例。计算机与 PLC Modbus TCP/IP 协议通信（针对国产和利时 LEC G3 PLC 程序，OMRON 用的是 FINS/TCP 协议），计算机为客户机，而 PLC 为服务端。PLC 程序在调用以太网功能块时自动生成，用户不必编程。计算机程序必须由用户编写。LEC G3 型 PLC 以太网通信采用 Modbus TCP 协议。它和 Modbus RTU 协议的区别在于：

第一，它没有从站地址这个概念，它寻址所依靠的是 IP 地址。

第二，在它的命令帧及数据帧中，没有 CRC 校验码。但是在帧之前要加入一些字节。具体内容见表 7-16。

表 7-16　Modbus TCP/IP 协议格式参考表

字节	说　明
0	节点 ID　一般为 0
1	节点 ID　一般为 0
2	协议 ID = 0　为 0
3	协议 ID = 0　为 0
4	从字节 6 到帧尾(数据帧)的字节数(高位) = 0(因为帧长度应该小于 256 个字节,因此,此位为 0)
5	从字节 6 到帧尾(数据帧)的字节数(低位)
6	地址识别,这一字节一般只是在 Modbus 桥接的时候用,在目前 LM3403 的接法,没有作用,一般为 0,模块不对这一位做判断
7	Modbus 功能码,诸如 0 × 05
8	后面就开始 Modbus 数据和地址,这个和 Modbus RTU 一致,只是不用校验

第三，它只能从指定输出区读数据。也只能对指定的输入区写数据。如要用 05 功能码写 LM3403 的第一个位（%IX4.0），其命令格式和表 7-17 所示：

表 7-17　写位数据命令格式举例

节点 ID		协议 ID		6～11:6 个字节		地址识别	功能码	起始地址		置位状态	原状态
0	1	2	3	4	5	6	7	8	9	10	11
00	00	00	00	00	06	00	05	00	00	FF	00

表 7-18 所示为和利时 LM3403 以太网模块寄存器区与 Modbus TCP 功能码对照参考表。

表 7-18　以太网模块寄存器区与 Modbus TCP 功能码对照表

区	读写	功　能　码		Modbus 地址与寄存器地址对应关系举例	
I	只写	位	0 × 05	%IX4.0	0
				%IX4.1	1
				%IX6.1	17
		字	0 × 06	%IW4	0
				%IW6	1
				%IW8	2
Q	只读	位	0 × 02	%QX 和%QW 与 I 区的对应关系相同	
		字	0 × 04		

注：（1）这里的 I 和 Q 区是 LM3403 设定的寄存器区。

（2）在进行网络通信时，PLC 编程软件通过串行接口可以同时连接 PLC，以监视和设置 PLC 各寄存器的状态和数值。

（3）位地址计算：如果位的地址为%IXx.y，那么，命令中的地址为（x－z）*8＋y。如字的地址为%IWm，那么，命令中的地址为（m－z）/2。这里的 z 为%IW 区开始地址。%QW 区计算方法与此类似。本表计算 I 地址，假设 z 为 4。计算 Q 地址，假设 z 为 2。

弄清 PLC 方情况，则编写计算机程序。图 7-61 所示为这个计算机程序表单。其上有发送报文文本框及接收报文文本框。有建立连接、发送命令及接收数据按钮。

作为客户机，计算机程序要点是：

在使用这些控件之前，要先把 Winsock 控件，从控件库中，调入到本工程 VB 的工具箱中。Protocol（套接字类型）属性设为 TCP，使用 TCP 协议。

用 connect 方法，请求连接。其参数是指定 PLC 的 IP 地址（PLC 配置时指定）及使用的通信端口（LEC G3 机协议规定为 502）。

生成命令，并用 SendData 方法发送命令。

用 GetData 方法，接收并显示数据。



图 7-61　计算机与以太网 PLC TCP IP 协议通信程序表单

具体程序为：

（1）建立连接程序。

```
Private Sub Command1_Click( )
  If Winsock1. State < > 0 Then Winsock1. Close
    Winsock1. RemoteHost = Text1. Text                    '确定服务器的主机名
    Winsock1. RemotePort = Val( Text2. Text)              '502
    Winsock1. Connect  ' "169. 254. 202. 1", 502 也可在此指定远程 IP 地址及端口号
    Do While Winsock1. State < > sckConnected: DoEvents   '等待连接
        If Winsock1. State = sckError Then GoTo skip
    Loop
Exit Sub
skip:
        MsgBox "连接出错 !"
End Sub
```

（2）生成并发送命令程序。

```
Private Sub Command2_Click( )
    If Winsock1. State = 7 Then
      Dim b(100) As Byte                                  ' 声明字节数组
      a $ = Text3. Text                                   ' 发送文本
      l = Len( a $)
      For k = 0 To 1 - 2 Step 2
        b( k \ 2) = Val( "&H" + Mid( a, k + 1, 2))
        ' 把可视的字符串转换为字节数组,两个 16 进制字符存于一个字节中
        Next k
      Winsock1. SendData b( )
    End If
End Sub
```

（3）接收并显示数据程序。

```
Private Sub Command3_Click( )
```

```
    If Winsock1. State < > 7 Then Exit Sub
    Dim rD As Variant
    Dim b( ) As Byte
    Winsock1. GetData rD
    l = Len( rD)
    b( ) = rD
    If 1 > 0 Then
      For i =0 To 1 -1          '根据接收数据字节数把不可视的数字转换可视字符
        aa = Hex( b( i) )         '把一个字节的 16 进制数转换为两个字节的字符
        If Len( aa) =1 Then aa ="0" + aa     '确保每个字节是两个字符
        a =a + aa                    '把单个字符组织成字符串
        Next i
        Text4. Text =a               '显示接收文本
    Else
        Text4. Text ="未读到数据 !"
    End If
    Winsock1. Close
  End Sub
```

该程序可用以读取 Q 区位或字，写 I 区的位或字。

从图 7-61 可知，它向 PLC 发送读取 PLC% QW4 字的命令，目的是读取% QW4 字的值，并予以显示。从图 7-61 可知，此命令已得到执行，并得到相应的回应。接收报文的最后 FF (65535)，即为此值。该程序还可以改变命令码，用以读取 Q 区其它位或字，也可用以写 I 区的位或字。

7.3.3　互联网通信编程

主要指通过同处于互联网上的计算机与 PLC，通过相互发送与接收电子邮件传送数据。图 7-62 所示为在互联网上计算机向 OMRON PLC 发送邮件的示意。与计算机间发送邮件一样，中间也是通过 SMTP 服务器。

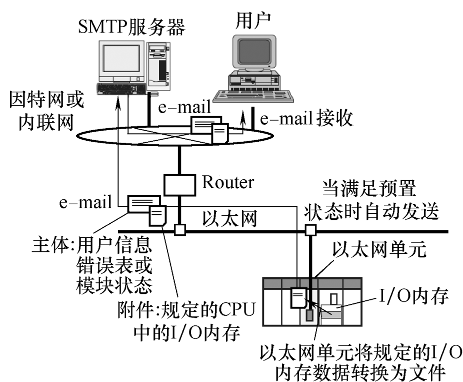


图 7-62　在互联网上计算机向 OMRON PLC 发送邮件示意

PLC 发送邮件由邮件头、邮件体及附件组成。而附件可以是由以太网模块自动生成的 I/O 内存数据文件，扩展名为 IOM（二进制）、TXT（文本）或 CSV（逗号隔开的数据文件）。也可是任意在 CPU 单元文件存储器中的文件。但每个邮件只能附加一个文件。

图 7-63 所示为加上附件 DATA0. CSV 的邮件传送情况。该附件含有 DM100 ~ DM119 200 个字的数据，每个字用逗号隔开。

PLC 什么时候发送邮件，由相应条件触发。此条件可以是用户设定的 CPU 单元 I/O 内存字段值大小或位的 ON/OFF 变化，也可是 PLC 工作状态变化，也可是定时触发。定时时间可在 10 分钟到 10 天之间设定。任一设定条件满足，都将向指定邮件地址发送邮件。

同样，计算机也可向指定邮件地址的 PLC 发送邮件。图 7-64 所示为 PLC 接收电子邮件的情况。经设

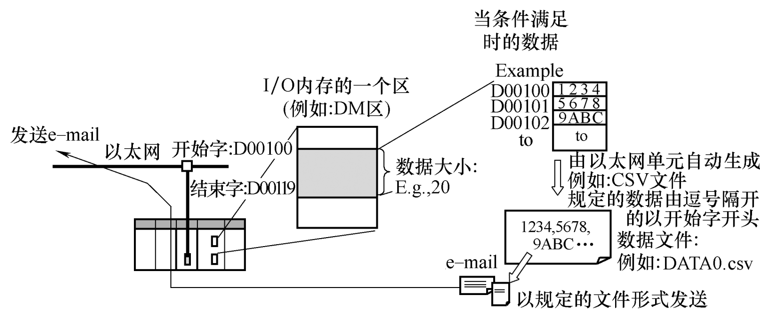


图 7-63　加上附件 DATA0. CSV 的邮件传送情况

定后 PLC 以太网模块会定时检查是否有邮件发来。如果有邮件，即可接收。接收完成后，将向对方发送回复邮件，以确认邮件已收到并回送相应的处理信息。

为了确保安全，可对收取的电子邮件作限定。如只能收取指定地址的邮件、限制对方邮件的命令、只能收取某种扩展名的文件等。如所收邮件不合上述条件，PLC 将不予处理。

图 7-65 所示为接收的含有 FileWrite（文件写）命令邮件格式。

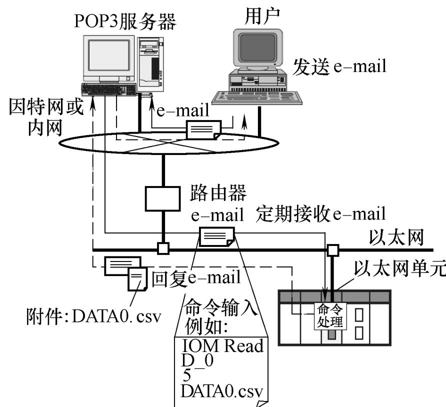


图 7-64　PLC 接收电子邮件的情况

对方地址	etn@omron.co.jp
标题	FileWrite
信件主体	Para1: MEMCARD ¥user Para2: Overwrite=OK #Overwrite OK
附件	Write.iom(98KB)

图 7-65　接收邮件格式

接收到此命令邮件成功执行后回复邮件的格式如图 7-66 所示。

对方地址	Command@omron.com
标题	Re:FileWrite
信件主体	Response Code:0000 Response Status: Normal end >Para1:MEMCARD ¥user >Para2:Overwrite=OK >#Overwrite OK >-----(Attached File was deleted)----
附件	Read.iom(98KB)

图 7-66　回复邮件格式

OMRON 公司的产品定义有多个接收的邮件命令。除了上述 FileWrite（文件写入）外，还有 FileRead（文件读取）、FileDelete（文件删除）、FileList（文件列表读取）、UMBackup（用户内存备份）、PARAM-Backup（参数区备份）、IOMWrite（I/O 内存写入）、IOMRead（I/O 内存读取）、ChangeMode（变更操作模式）、ErrorLogRead（错误记录读取）、ErrorLogClear（错误记录清除）、MailLogRead（邮件记录读取）、MailLogClear（邮件记录清除）、Test（邮件检查）、FinsSend（FINS 命令发送）等。这些命令还都有各的发送与回复格式。但回复码（Respone Code）总是要依据接收邮件的情况自动确定。其含义见表 7-19。

要指出的是，由于邮件的数据量较大，所以，传送的时间是较长的。表 7-20 所示为 CJ1H 型 PLC 不同长度邮件在 PLC 处编程及监控状态下的预计接收时间。

表 7-19　接收邮件回复代码表

响 应 码	含　义	响 应 码	含　义
0000	正常执行	F301	译码错
F101	信件太大	F302	附件非法
F102	地址错	F303	没有附件
F103	非法命令	F304	文件被保护
F104	命令执行受保护	F305	文件太大
F105	非法标题	F2FF	其它错误
F201	非法参数		

表 7-20　CJ1H 型 PLC 不同长度邮件接收时间

命　令	数据大小	CPU	
		编程	运行
		—	10ms 循环时间
FileWrite	1kB	0.2s	0.4s
	10kB	0.9s	2.6s
	100kB	9.0s	25.7s
	1MB	90.5s	302.8s
FileRead	1kB	0.1s	0.3s
	10kB	0.4s	1.8s
	100kB	4.0s	17.8s
	1MB	48.4s	272.0s
IOMWrite	1word	0.1s	0.1s
	6000words	0.1s	0.2s
IOMRead	1word	0.1s	0.1s
	6000words	0.1s	0.2s

从上介绍可知，这个电子邮件通信的功能是很强的。使用它可使上位计算机在互联网所覆盖范围内与 PLC 通信，实现与其数据交换及对其实施操作。但这也只是将 OMRON 100Base-TX 以太网接入互联网后才有可能。

7.3.4　公网平台通信编程

1. 固定电话网络通信编程

计算机、PLC 串行接口，使用 Modem（调制解调器），可通过市话系统通信。凡是电话能到达的地方都可通信。

通信前，如 7.1 节介绍的，双方 Modem 都要先做好设置。之后，计算机执行呼叫的代码，即 atdt + PLC 方电话号码 + 回车。一旦呼叫成功，将返回相应信号，双方 Modem 的 CD 指示灯都亮。这时，计算机与 PLC 通信，虽经过 Modem、电话局……但仍如同直接连线，可方便地进行。

有的厂商生产的 PLC，如西门子 S7-200，还有 Modem 模块，在模块上做好设定。如经计算机成功呼叫，则建立了连接也可通信。

这样的模块不像通用的调制解调器，而是一个智能扩展模块，不占用 CPU 的通信口。它有密码保护及回拨功能。可通过模块上的旋转开关，实现从 300baud 到 33.6kbaud 的自动波特率选择。是用脉冲还是用语音拨号，也可选择。

2. 移动电话网络通信编程

移动电话在我国发展很快。目前装机量已超过固定电话。所以，利用它进行 PLC 与计算机通信或与个人手机互发短信是很方便的。

为此，要有 2 台 GSM 的 Modem，如 BM2403A。还要有 2 张手机的 SIM 卡。系统配置如图 7-67 所示。

此外，还要对串行接口特性、通信模式进行设置。把 PLC 串行接口要设置为无协议方式，特性与 Modem 一致。并编写调用 TXD 指令程序。

运行 PLC 程序后，当执行 TxD 指令条件具备，则自动发送 AT 指令给 GSM Modem，Modem 再将设定好的信息以短信的方式发送给用户手机、计算机。只要在中国移动通信的网络覆盖范围之内，就可收到此短信。

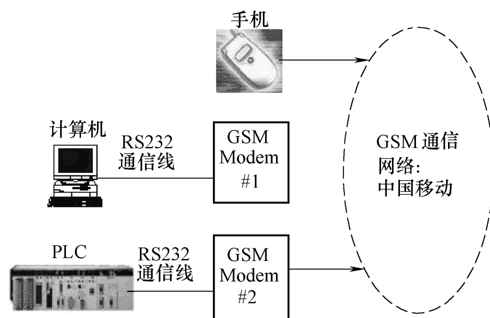


图 7-67 PLC 移动通信配置

如发送“OK”。其 AT 命令为：

AT + CMGS = “13912345678”（报头及手机号码）回车（结束符）OK（发送信息）发送符

以上为字符，实际要转换为 ASCII，并要预先存放在 TxD 指令的源字中。如在它的开始字为 DM100 中，则对应的在 DM100 ~ DM112 中的值（十六进制）分别为：

DM100（4154，即字符 AT），DM101（2B，即字符 + C），DM102（4D47，即字符 MG），DM103（533D，即字符 S = ），DM104（2231，即字符 1），DM105（3339，即字符 39），DM106（3132，即字符 12），DM107（3334，即字符 34），DM108（3536，即字符 56），DM109（3738，即字符 78），DM110（220D，即字符回车），DM111（4F4B，即字符 OK），DM112（1A00，即字符！Null）。

3. 无线网络通信编程

如果图 7-67 中的 Modem 换为无线 Modem，通过专用无线电台传送信号，就成了无线网络。威海市自来水公司调度中心计算机与全市 20 多个供水站 PLC 之间就是这样通信的。各个供水站用 PLC 控制进、出水阀门的开闭。并实时检测与记录进、出水流量、压力。中心计算机定时与 PLC 通信，采集各水站的这些数据，并通过 PLC 控制各水站工作。只是这里的无线网络是专用的。不是公网。

7.3.5 工具软件通信编程

工具软件多为 PLC 厂商提供，兼用或专用于计算机与 PLC 通信。最常用的就是编程软件。此外，还有 OPC 服务器、ActiveX 控件及 API 函数。一般讲，工具软件在任意平台上都可运行。

1. 编程软件

OMRON 公司的 CX-Programmer 除了用于编程，还是最基本的通信工具软件，可在所有 OMRON 网络平台上，实现计算机与 OMRON 所有 PLC 联机通信。所以，要是哪个网络“叫不通、连不上”，用它进行测试是最好的选择。如果 PLC 通信接口设定出现问题，无法与计算机通信，最好的方法也是对 CPU 模块上的硬件开关恢复为默认设定（不同机型开关有所不同，可参阅有关说明书），再用编程软件按默认设定即可通信。

此外，CX-one 中的 Integrator 是网络配置工具，用以实现网络通信及测试。CX-one 中的

其它工具软件多也可用以与 PLC 通信，只是使用它的技巧要高些。

2. ActiveX 控件

很多 PLC 厂家都开发有针对可视化编程软件使用的、针对自身 PLC 串行接口或网络模块的通讯控件（Active X 控件），如 OMRON CXP-One 软件，在计算机上安装后，这些控件将加载到编程软件平台上。图 7-68 所示为在 VB 控件视窗上显示的 OMRON PLC 的控件。

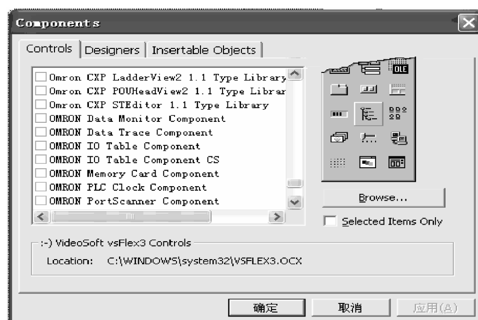


图 7-68　OMRON PLC 控件

图 7-69a 所示为仅加载一个 Command1 及一个 DataMonitor1 控件使用的窗口。加载后用鼠标右键点击该控件图标，将弹出一个菜单，从中选定特性，则显示如图 7-69b 所示的“Property Pages”（属性页）。在该页面，可设定连接的 PLC 及通信网络。本例连接 CP1H 型 PLC，使用 USB 接口。

这时，如果点击左上角“通信”，则转为显示通信页面。这时再点击“测试通信”按钮，则在窗口右方显示测试信息。这里显示“已连接”，说明连接成功。



DataMonitor1 控件

a)



b)



c)

图 7-69　DataMonitor1 控件使用

a）FORM1 窗口　b）特性窗口 1

c）特性窗口 2

在本例中，Command1_Click（）过程，仅写 DataMonitor1. Show 一个语句。其含义是打开 CX-one 内存窗口。所以，如果本程序运行点击 Command1 按钮，将弹出如第 2 章的图 2-92 所示内存窗口。对其使用与在 CX-Programmer 编程软件中一样。

用厂家控件的好处是，可以不弄清通信协议就可编写通信程序。然而，这里的控件大多数因为没有交费、没有驱动，无法实际使用。

3. OPC（OLE for Process Control）

OPC 是微软处理程序间通信和数据交换的新技术。使用 OPC 可把通信程序与应用程序分开。通信程序由 PLC 厂家提供，用于与 PLC 通信，并用作 OPC 服务器，用一组组接口（interface）为客户提供服务。而应用程序由 PLC 用户编写，作为 OPC 的客户，通过接口接收服务，使 OPC 客户间接实现了对 PLC 监控与数据采集。

OPC 接口由 3 类对象组成，相当于 3 种层次上的接口：服务器对象（Server）、组对象（Group）和数据项（Item）。

（1）服务器对象（Server）

拥有服务器的所有信息，同时也是组对象（Group）的容器，一个服务器对应于一个 OPC Server，即一种设备的驱动程序。在一个 Server 中，可以有若干个组。操作系统用 CLSID，即 128 位长的标识码识别。在每一这类文件安装时，由操作系统向其指定的惟一标识码。

（2）组对象（Group）

是应用程序组织数据的一个单位。客户可对之进行读写，还可设置客户端的数据更新速率。当服务器缓冲区内数据发生改变时，OPC 将向客户发出通知，客户得到通知后再进行必要的处理，而无需浪费大量的时间进行查询。OPC 规范定义了两种组对象：公共组（Public 或称全局组）和局部组（Local 或称局域组、私有组）。公共组由多个客户共有，局部组只隶属于一个 OPC 客户。全局组对所有连接在服务器上的应用程序都有效，而局域组只能对建立它的 Client 有效。一般说来，客户和服务器的一对连接只需要定义一个组对象。在一个组中，可以有若干个项。

（3）数据项（Item）

是读写数据的最小逻辑单位，一个项与一个具体的位号相连。项隶属于某一个组。是服务器端定义的对象，通常指向设备的一个寄存器单元。OPC 客户对设备寄存器的操作都是通过其数据项来完成的。通过定义数据项，OPC 规范尽可能的隐藏了设备的特殊信息，也使 OPC 服务器的通用性大大增强。OPC 数据项并不提供对外接口，客户不能直接对之进行操作，所有操作都是通过组对象进行的。

使用 OPC 技术是趋势，OMRON 也不例外，使用 SysmacOPC。事实上，当 CX-one 软件安装后，在 OMRON 项目之下，就有“CX-server”项。之下还有“DDE 管理器”、“驱动管理器”、“输入输出工具”、“性能监视器”等项目。可用于 OPC 服务。只是它只是外壳，付之使用还得有另加付费的软件激活。

对于 OPC，用户操作数据项的一般步骤为：

1）通过服务器对象接口枚举服务器端定义的所有数据项。

2）将要操作的数据项加入客户定义的组对象中。

3）通过组对象对数据项进行读写等操作。

每个数据项的数据结构包括三个成员变量：即数据值、数据质量和时间戳。数据值是以变量类型表示的。

这时，如用 VB 编程，则要先在 VB 开发环境平台上，加载 OPC 类。还要定义与这些类有关的窗口全局变量。以便在程序中使用这些变量。再次是，建立 OPC 连接，加入项目组。这些可用初始化程序实现。

4. FinsGateway

FinsGateway 现在版本是 FinsGateway2003。是 OMRON 公司 FINS 协议的驱动程序。通过该驱动，上位机可以通过各层网络（包括网络互连）来访问网上的 PLC。若采用运行版（Run-time 版），上位组态软件可直接通过 FinsGateWay 的驱动，方便地与 PLC 进行通信。若采用开发版，则该软件提供 Sysmac Compolet，即 VB/VC 控件，同样也可以为自己开发的程序提供驱动。

FinsGateway 有两种通信服务。一是 FINS 信息服务通信，使用网络通信命令。计算机发送，PLC 响应，总是成对的。二是数据链接通信，用内存共享。在计算机方，参与共享内存称之为事件内存（EventMemory），可与 PLC 的 DM 区进行数据链接通信。事件内存还可作为 FINS 服务器，被其它应用程序访问。

图 7-70a 所示为 FINS 信息服务通信示意。它的数据交换是靠发送通信命令及接收响应实现。计算机接收到数据存于事件内存中。图 7-70b 所示为事件内存与 PLC DM 数据链接通信示意。它的数据交换是自动实现的。

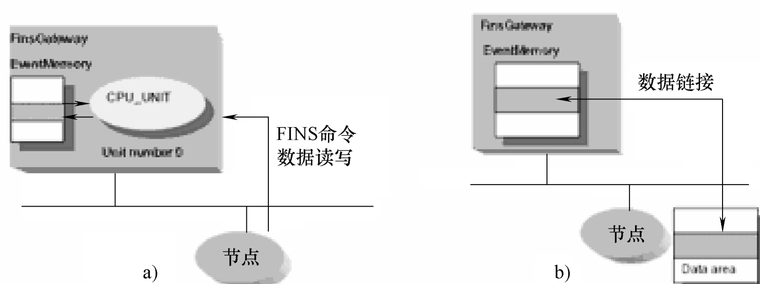


图 7-70　数据链接示意
a）FINS 命令数据读写　b）数据链接

FinsGateway 还有网络中继功能，可跨网络通信。还有套接字代理服务器功能，允许被 TCP/IP 网络上程序访问。此外，OMRON 的还有 PLC Reporter32 数据收集软件，安装后在 EXCEL 表格中直接设定，可把读取 DM 区的数据，转为 Excel 表格显示。

5. API 函数

有的厂商，如西门子，不提供网络或串行接口通信协议，只提供它自己开发的通信用 API 函数。这些函数是在安装它的 Prodave 通信软件后加载给 Windows 的。使用这些 API 函数，即使不清楚它的通信协议，也可编写使用串行接口及 Profibus 网络的通信程序。

Prodave 通信软件提供的 API 函数很多，如：load_tool（）接口设定及打开、unload_tool（）接口关闭、d_field_read（）读 DB 模块、d_field_write（）写 DB 模块等。还有很多其它软器件的读、写函数。这些函数可用于 S7 各个机型，可用于 MPI 网，也可用于 Profibus 网。

正如使用 Windows 的 API 函数一样，如 VC 使用，先写好头文件调用语句就可以了。而如果 VB 使用，必须先对函数进行声明。以下就是打开口（load_tool）与关闭口（unload_tool）的函数声明。

```
Declare Function load_tool Lib " w95_s7. dll"（ByVal nr As Byte，ByVal dev As String，adr As plcadrtype）As Long
Declare Function unload_tool Lib " w95_s7. dll"（）As Long
```

7.3.6　PLC 方编程

如果 PLC 与计算机为被动通信或协议通信，PLC 方基本上可不用编写程序。但为了提高程序效率与性能，多数还是要编写一些准备数据及使用数据程序。有时还要执行初始化程序来进行通信口设定。如果为主动通信，PLC 方必须编程。

1. 被动通信

（1）数据准备程序。最好把上位机要读的数据作些归拢，集中在若干连续的字寄存器中。这样一来，当上位机读取时，一个命令即可读走。如果数据较分散，则要用多个命令多次读。这既增加通信时间，又增加上位机编程的工作量。

（2）数据使用程序。首先，为了让计算机能向 PLC 写数据，应使 PLC 处于监控工作方式（对 Host Link 协议）。可用 PLC 起动方式设定，或用计算机写 PLC 处于监控状态实现。此外，最好也把上位机要写的数据作些归拢，集中在若干连续的字中写。这样，上位机向下

传数据一个命令就可以了。不然如果数据较分散，则要用多个命令多次写。只是要注意，数据使用后要用程序及时还原。

图 7-71 所示为一个“位数据”使用程序。这里触点 A 置位，受计算机写控制。

从图知，这个“A”与“按钮按”的作用相同。即“工作”既可由“按钮按”控制，也可由“A”控制。而紧接着程序为用“P_Off”写 A，使“A”复位（复原程序）。其含义也与实际的“按钮按”总是按后及时松开一样。为了便于理解与修改，也可把所有的复原程序，集中放在主程序的最后面。如图用值 0 赋值给所有被写位。

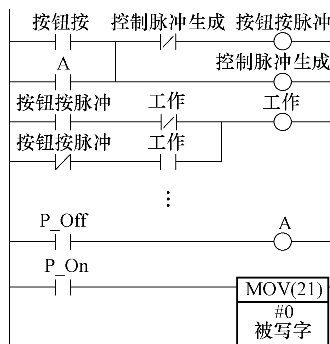


图 7-71 数据使用程序

写字数据比较好处理。可直接写，也可写在一个连续的数据区，然后用传送指令传送给数据的使用区。

2. 主动通信

主动通信是 PLC 发起的。PLC 根据控制状态或采集到的数据情况，主动给上位机发送数据，等待计算机回应。当计算机接收到这数据，再按约定向 PLC 回复数据回应命令。PLC 再对回应进行判断，以进行下一步处理。图 7-72 所示为 PLC 主动通信的一个例子程序。

从图知，当 9.01 ON（要进行某个控制）时，PLC 向串行接口发送一组数据。注意，这里的 TXD 指令为微分执行，即仅发一次数据。控制字 0，说明用 RS-232 端口发数据。第 3 个操作数为#4，说明发 DM10、11 中 4 个字节数据。从程序知，这 4 个字节的内容为“1112AAAA”。

计算机则不断的读串行接口，或设定串行接口中断工作。一旦收到此数据，经判断、确认，如按约定发向 PLC 的 DM1 写“ABCD”的通信命令。如 PLC 接受了这个写操作（注意，C 系列机可设为 Host Link 模式，则不须编程，系统自行实现），从图的梯级 2 知，其比较结果使 P_EQ（相等标志）ON，则使 8.01 置位，并自保持，程序进入下一步操作。且使 9.01、DM1 复位，为以后通信应答作准备。

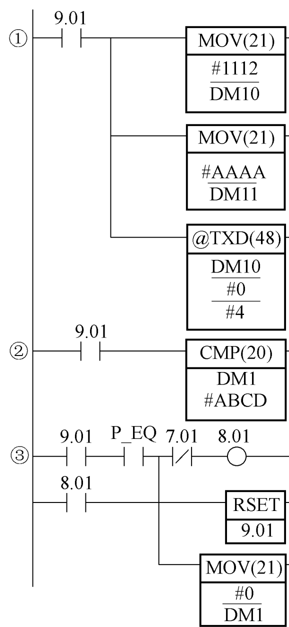


图 7-72 主动通信程序

如果程序再细一些，还可考虑加定时控制，一旦长时间得不到计算机的回应，便再发通信数据，或报警。还可再作别的比较，如 DM1 为其它某个数，则相应其它分支操作，等等。

提示：对 CP、CJ 系列 PLC，主动通信最好用 FINS 协议。否则不是处于 Host Link 模式，还不能接受上位机通信命令。那样，要使用主动通信，则双方只好都得如同 PLC 之间自由协议通信一样处理。

7.4 PLC 与计算机通信编程（二）

关键词：操作系统、编程软件、应用软件、组态软件

本节将讨论计算机用组态软件与 PLC 通信编程

7.4.1　组态软件概念

计算机软件的种类很多。最基本的是操作系统，如不同版本的 Windows 就是操作系统。它用以管理个人计算机的硬件与其它软件，是计算机最重要、最基本的软件。没有操作系统，计算机将无法工作，别的软件也无法运行。

除了操作系统，计算机还有编程软件、应用软件。如 VB、VC 等就是编程软件。它为应用软件开发提供平台。而用 VB 编写的计算机与 PLC 通信程序则是应用软件。

尽管编程软件已提供了可视化界面。但用它开发应用软件还是较麻烦。开发周期也较长。不适合于系统控制工程师，只适合计算机编程人员。

所以我们都期盼着有这样的软件，它介乎编程软件与应用软件之间，是专业公司在编程软件平台上开发的，商品化的通用“应用软件的半成品”。利用它，系统控制工程师再作些的不太复杂的再开发，就可成为一些数据采集与过程监视及控制的专用的应用软件。这“介乎中间”的软件，就是这里将要介绍的组态软件。

大约在 20 世纪 80 年代中期，组态软件在美国应运就诞生了。它在 80 年代末 90 年代初进入中国。目前世界上有不少专业厂商，包括专业软件公司和硬件制造厂商，都生产和提供这种组态软件。

在当今中国市场上，组态软件产品按厂商划分大致有三类，国外专业软件厂商提供的产品，国内外硬件或系统厂商提供的产品及国内自行开发的产品。但市场的大部分份额仍被国外几个组态软件占据，如 FIX、Intouch 等。

不管是国内的还是国外的组态软件，作为商业产品，一般都提供了友好的用户使用界面，还有变量库、图库、控件库以及脚本语言。同时，还都还提供（收费或免费的）种种与硬件通信的驱动程序。所以，有了它，用户可简便、灵活地设计画面，定义变量及调用驱动程序，基本上不用编写程序代码，或编写篇幅不大的脚本，就可组态成自己的应用软件。而实现的功能却与用编程软件开发的应用完全相当，而且一般还都具有更漂亮的画面。

随着计算机软件的发展，特别是组态软件的广泛应用，组态软件发展也很快：

组态软件正在由单一的人机界面，向数据处理机的方向发展。管理的数据量越来越大。大型数据库技术也在组态软件中得到应用。很多组态软件，如力控、Internation、Wonderware，还开发有自身的实时数据库。它既是组态软件的一部分，又是独立的软件产品。

随着计算机网络的飞速发展，也出现了分布式、网络化、大型的组态软件。有的能直接支持 Internet 远程访问。

组态软件有的还可冗余备置，可热备工作。一旦一个计算机出现故障，备份的仍可进行监控。为了适应高可靠工作行业，如电力工业的要求，还出现种种专业版，如电力版。

商用软件技术，如动画技术、分布式运算等，也已逐渐在组态软件上得到应用。

同时，组态软件还向小型化发展，主要用于嵌入式计算机。

早期组态软件的主要问题是：国外产品价格太高，用不起；国内的价格很低，但不稳定，不敢用。再就是组态软件速度低，不够灵活，很难完全适合所有工程实际的要求。所以，很多人宁可自己用 VB、VC 这样编程软件编写人机界面程序而不用组态软件。

而当今情况已有了很大改变。国外软件价格在降低，国内的软件质量在提高。随着计算机工作速度的不断提高，速度低已不再是尖锐的问题。组态软件也改善了自身的脚本语言及

增加了多种程序接口，灵活性也有了很大提高。几乎大部分用编程软件所编程序的功能，用组态软件编的程序，也都能实现。

再加上组态软件自身的固有优点：

易学习，学半个月、一个月就可上手，而 VB、VC 没有一年半载很难入门；

易开发，用组态软件开发应用，一个中等难度的一般工程，半个月可以了，VB、VC 少的一两个月，长的就更难说；

易维护，组态软件开发的应用、维护、修改很容易，而 VB、VC 开发的应用，换一个人去读懂代码都不易，改就更难。编程的人“跳槽”往往使“老板”为难也与此有关；

界面好，组态软件是专业厂家作为商品提供的，界面都比较美观，所以，用它开发的应用画面都可做得比较美观；

接口多，组态软件接口多，上可接多种数据库和管理网络，如 ERP 系统等，下可接多种现场设备，可发挥承上启下的作用。

此外，随着计算机的发展及信息化的推进，组态软件的概念也在改变，功能也在扩展。已由单纯地用于开发人机界面（HMI）或监控数据采集系统（SCADA）的工具，发展成自带数据库、自带工具包的综合平台。用组态软件既可开发监控及数据采集应用，又可开发种种数据处理应用、网络服务应用，以至于企业管理系统应用。

所以，在自动化系统中，组态软件应用越来越广，作用越来越大，产品越来越多，前景也越来越看好。

7.4.2 组态软件简介

目前进入市场的组态软件有一二十个主要品牌。互联网上都有这些产品的介绍。以下择其主要者，分国外、国内两部分，作一简单介绍。

1. 国外组态软件

（1）InTouch。Wonderware 公司开发的组态软件。Wonderware 公司创建于 1987 年 4 月。开创了组态软件的先河。

InTouch 应用范围广泛，已有 20 多万套软件运行在全球 25000 多个工厂中。其市场包括食品加工、半导体及电子业、石油及天然气、汽车、化工、制药、纸浆与造纸、运输、智能楼宇、水电公用事业和其它各专业领域。

InTouch 内涵丰富，组件也很多，版本很多，并不断创新，现已升级到 8.0。它有“抗版本不一致性”的特点，即所有部件同时升级。这可保证整套系统即插即用，配合默契。

InTouch 版本升级后，客户在早期 InTouch 版本上开发的应用可自动地移植到新版本上。这为用户应用更新提供了很大方便。

InTouch 只是它的统称。其中 Wonderware FactorySuite 是它的工厂套件，是集成的，基于组件的 HMI 系统。

InTouch 软件开发环境通用，但体系结构灵活，可适应不同的自动化应用场合。它可以是：

1）单机使用。应用安装在单个电脑上。对于不需要多个操作员站来观察和控制同一个工业过程时，常是这么使用。它的每个节点都是完全的系统，运行不依赖于任何其它电脑。而且，这些系统也可联网。

2）客户机/服务器。应用安装在一个客户机/服务器环境中。这种方法有利于节省软件维护和管理。它有几种不同的情况。

3）标记服务器配置。在进行这种配置时，可以选定一台电脑作为标记服务器，也可以选定多台电脑作为标记服务器。标记服务器存储标记名（tagname）词典（在 InTouch 应用中使用的全部标记），记录历

史事件、运行 QuickScript（脚本）、作为一个报警设备、并连接输入/输出（I/O）数据。运行在客户机节点（操作员站）上的应用连接到标记服务器，可显示信息。

4）动态网络应用开发（NAD）。动态网络应用开发（NAD）可以通过网络服务器来集中维护 InTouch 应用。在每个客户机节点上，建立主应用的一个本地副本。这种方法具有冗余特点，如果服务器不可用，客户机节点使用应用的本地副本仍能正常工作。当服务器恢复正常后，重新连接完全是透明和无缝的。

NAD 的另一个强大的特点是，用户可以在客户机节点上接收 InTouch 应用的变化，而不必停止 InTouch 应用的运行。当应用被改变时，系统为操作员提供警报，此时操作员可以在方便时接收变动。接收变动后，只有发生变化的应用程序组件会被下载到客户机节点上并更新。如果操作员选择不接收配置变化，那么在下一次系统重新起动时可以下载最新的应用程序。因此，操作员可以始终使用当前的应用，并且可以在任何时候更新运行的应用，不会导致系统停工或者过程的可视化内容的丢失。

5）终端服务。终端服务体系机构允许对多种操作系统进行集中部署、软件维护和管理、硬件的重用、高级安全和支持客户机，包括 Windows CE、嵌入的 Windows NT、Windows for Workgroups 3.11、95、98 和 NT3.51/4.0、2000、XP、Linux 和 Unix 操作系统。另外，客户可以使用瘦客户机终端把视图延伸到他们的过程中。瘦客户机终端可以与常规的计算机节点一起使用，为应用提供额外的低成本视图，或者替换指示设备，例如图表记录仪或温度控制器。另外，InTouch 应用的终端业务还可以在个人数字助理（PDA）上运行。这样，用户可以在自由移动的同时仍拥有对应用的恒定视图和控制。

6）InTouch 视图。实现 FactorySuite 工业应用服务器的系统，可以使用 InTouch 为过程提供视图。工业应用服务器可以极大地减少在一个工厂或跨多个工厂维护和部署大型系统所需的工程工作和时间。

7）其它的分布式系统特性。InTouch 还提供几个附加的特性，以支持对分布式环境更好地进行应用设计和控制。

InTouch 还有如下特点：

1）InTouch WindowMaker 中，可使用多种工具开发图形。这些包括标准的图形组件、位图图像、ActiveX控件，以及符号工厂（Symbol Factory）。Symbol Factory 是一个高级图形库，它包含数以千计的预先配置的工业图形。所有这些工具都非常易于使用和直观，因此，用户可以快速开发和部署可视化应用。

2）InTouch 脚本（QuickScript）编辑器，可以扩展和定制 InTouch 应用，以满足特定的系统需求。可以根据众多的参数配置脚本，例如特定的工艺条件、数据变化、应用事件、窗口事件、键盘敲击事件、ActiveX 事件等等。QuickScript 环境还支持 QuickFunctI/Ons，允许用户开发一个可重用的脚本库，从而简化应用，减少初始工程和应用维护时间，简化应用部署。

QuickScript 编辑器简单易用，允许制定所有的应用过程。在生成脚本时，可以在带有常用的表达式和结构，例如 for-next 和 if-then else，的按钮上点按。高级功能（例如数学函数、字符串转换函数等等）可以通过向导调用，在调用这些高级功能时，系统会提示输入必需的参数，保证函数语法的正确性。

QuickScripts 和 QuickFunctions 支持使用局部变量存储临时结果和建立带有中间脚本值的复杂计算。在 QuickScripts 和 QuickFunctions 中使用局部变量不会占用许可的标记名数量。

QuickScript 的内嵌验证引擎，允许在部署脚本之前进行验证，防止运行时出错。另外，还可以在脚本编辑器中，编写和编辑脚本，或者从其它应用剪贴，有助于重用和节省设计时间。

3）InTouch 软件包括一个分布式的历史趋势系统，该系统动态地为每个趋势图表笔指定一个特定的历史文件数据源。这样，就可在同一个趋势图中观察本地 InTouch 历史和 IndustrialSQL Server 上的历史信息。分布式的历史趋势能力，可在一个屏幕上快速分析历史信息，在节省时间的同时，能够更好地分析多个变量。

4）DRC（动态分辨率转换）能力，使用户能够以一个屏幕分辨率开发应用，然后以另一个分辨率运行应用，不会影响原来的应用。应用还可以在用户定义的分辨率下运行，而不一定使用显示分辨率。另外，DRC 允许客户在一个应用中利用多个监视器的优点，不必担心窗口出现的位置。通过一次创建应用，不需要重新设计、复制或修改原来的应用，就可以在任何地方和任何大小的显示设备上部署该应用。

5）InTouch HMI 可以为分布式的历史和报警系统同时提供服务，允许以当地时间查看数值。这是非常重要的，因为当应用的规模变大到跨越多个物理区域时，它消除了事件发生的混乱性。

6）InTouch 利用了通信技术中的标准，并且把它们与微软技术相结合，为应用开发者提供了更开放的、易用的开发环境。InTouch 支持用户使用最新的设备通信协议，包括 Wonderware 的 SuiteLink 协议、OPC、标准 DDE、fastDDE 和 NetDDE。另外，InTouch 还是一个容器。它允许 InTouch 用户安装第三方 ActiveX 控件，并可使用点击鼠标的方式配置这些控件。把它们应用到任何应用程序窗口中，根本不需要编程。

7）InTouch 8.0 是冠有“Designed for Windows XP” Microsoft 证书的 HMI。故 InTouch 应用可以无缝地安装和运行在 Windows XP 平台上。8.0 版本还利用了新的 XP 特性，因此，系统管理员可以方便地从 XP 平台安装和删除驱动程序。

8）InTouch 的标记名（TagName）浏览器允许用户从任何 FactorySuite 应用（例如另一个 InTouch 结点等）选择标记名和标记名字段。这种能力支持在应用程序之间快速配置，为开发人员和同步标记名，实现简化管理和维护节省了大量的时间。

9）InTouch 的 FactoryFocus 图形监控软件还提供了 InTouch 8.0 HMI 运行版中的只看功能。它为经理和监管人员提供了实时观察连续的 HMI 应用进程的能力。使用只看功能，由于不能改变任何数据，所以系统安全得到了增强。

10）InTouch 的标记名交叉引用功能允许用户分析标记名、超级标记（SuperTag）和远程标记引用的使用情况。它指出特定的标记名或引用被使用的窗口或 QuickScript。为了方便起见，标记名交叉引用窗口可以始终打开在 WindowMaker 编辑程序上，而开发人员可以执行其它的任务。它还允许直接观察其中包含标记名的 QuickScript 或 QuickFunctI/On。

11）FactorySuite 的历史数据库，InSQL Server，包含生产信息。InSQL 为定义存储数据的获取源提供了灵活性。完整的生产数据可以实时地、自动地从工业标准的 OPC 服务器获得，也可以从 I/O 服务器所支持的 600 多种硬件设备上获得，还可以从众多的 InTouch 和 InControl 节点上获得。

对于非实时数据或存在于生产网络之外的数据，InSQL 支持从 .csv 格式的文件中导入数据。另外，InSQL 还支持通过标准的 SQL INSERT 和 UPDATE 语句输入数据。

不论数据源在哪里，数据源的状态如何，或者数据输入的时间是什么时候，所有数据都被集成到一个统一的存储地点，用户可以方便地使用按时间顺序排序的数据。这种集成提供了一种无缝的、方便的历史数据回取方法，从任何客户机上可以使用配置、报警、事件和概要信息。

InTouch 还有很多组件及相应特点，如：

InControl——用于基于 Windows NT 的机器和过程控制；

InTrack——资源跟踪工具，用于资源管理；

InBatch——批量生产管理系统，用于柔性批量生产；

I/O Servers——用于与绝大多数的控制与数据采集设备通信；

SuiteVoyager——用于远程查看的 Internet/Intranet 工具

ActiveFactory——IndustrialSQL 的数据分析工具

SCADAAlarm——企业级的电话/报警系统

T Analyst——停机跟踪和生产监视系统

ArchestrA——工业自动化软件的开发平台

Industrial Application Server——采用 ArchestrA 体系的 FactorySuite

SCADASuite——实时信息管理和监视控制的集成套件

等等。

总之，InTouch 功能很强，性能也很高。是当具有较高国际水平的专业组态软件产品。

（2）FIX、iFIX。Internation 公司的产品。Internation 公司也有 20 余年的组态软件开发历史，也是组态

软件产品的大厂家。现被 GE Fanuc 收购，成为它的一员。在全球“财富”500 强工业企业中，有 85% 的企业使用了它的产品。

它的产品适用各种 PLC，例如支持 OMRON、西门子 200/300/400，三菱 A 系列，Fx 系列；莫迪康，GE 等。支持各种工控板卡，RS_232，RS_485，honywell 公司 R_150，R_160，S9000，各种智能变送器．也可用于工业控制管理网络。

FIX、iFIX 是一个软件系统。最基本的功能是数据采集和数据处理。FIX、iFIX 提供了大量 I/O 驱动器。这些驱动器具有查错、报告、恢复、内置数据报告以及支持冗余通信。FIX、iFIX 软件还支持 DDE 服务器。Fix、iFIX 提供 I/O 设备的驱动程序。

FIX、iFIX 具有强大的 ODBC 技术，支持 Oracle、Sybase、FoxPro 等关系数据库。

它组件也很多，有：

iFIX——分布式的人机界面和过程可视化解决方案；

iDownTime——设备故障诊断专家系统；

infoAgent——基于 Web 的生产数据可视化分析工具；

iWorkInstruction——iBatch 的 S-88 配方工作流指令组件；

iWebServer——基于 Web 的人机界面解决方案；

iHistorian——企业实时历史数据库平台，以极高的速度采集、归档，并发布海量实时的现场信息，每一台服务器可同时采集 100，000 数据点；

iBatch——批次过程的监视与控制解决方案；

等等。

这些组件为生成 HMI，SCADA，SoftLogic，Batch 和 Internet 等一系列完整的工业自动化监控及行业自动化解决方案，提供基于组件对象技术的开发平台与工具。

其中的 iWebServer 主要特点是：

它是一个瘦客户端应用，运行在 web 服务器上，允许远程用户使用台式或便携电脑，以只读方式访问工厂的过程，而无需安装特殊的软件、驱动程序或客户应用程序。故远程浏览工厂信息的过程非常简单：登录到 Internet，访问过程画面的地址，就可实时地浏览 iFIX 过程画面。

还具有报警历史特性，通过浏览器浏览报警，从而可快速响应一个报警状态。使用 Internation 新的数据回放功能，可回查系统中的历史数据，以理解出现的问题，并进行远程处理。

可设置权限，依不同权限，提供不同的访问画面。并可决定使用 iWebServer 浏览 iFIX 过程画面的用户数量。还可快速方便地添加 iWebServer 节点。

介于厂级和 Internet 或 Intranet 之间，隔离了 SCADA 节点，防止未经授权的访问。不必担心重复请求访问过程画面时 SCADA 节点的延迟问题，因为 iWebServer 通过 web 服务器管理所有的请求，从而保证整个 SCADA 顺利运行。

等等。

此外，FIX、iFIX 还具有几个“易”的特点：

1）易学习。可用动画式在线示范，在学习创建定制显示画面的同时，迅速掌握系统开发的全部要点。

2）易开发。在集成的工作环境里，可快速方便地建立应用系统，增加、修改、删除和查看 I/O 信息。还可访问其它系统开发应用程序。如历史计划、历史显示和配方生成器。

利用预制的 Dynamo 图形库创造画面。这些图形库包括了泵、阀门、面板、仪表、管道、按钮和标记框。使用时，只需简单地从图形库中将图例拖曳进画面。组态操作极其简单，因为 Dynamo 会提示所有必要信息。

通过快速查看功能，可以立即验证画面是否会按你希望的样子工作。

不必重新起动系统或丢失任何有价值的数据，可以在线地构造和增强您的应用。

脚本能力也很强，可以建立简单或复杂的命令序列。

3）易使用。利用直观的、面向对象的图形用户界面（GUT），可设计出显示画面。可从 Dynamo 出发建造画面，也可以利用绘图工具和无限的颜色构造对象。这些对象可基于过程值而缩放、填充、旋转、变色和移动。

利用历史趋势，可以收集、储存和显示实时过程数据。归档的数据可以用于形成文件、整理报告及比较分析。并可确信所得到的过程数据是完全精确的。

能制作精确的、定制化的报告。通过双向的 DDE 接口，可与任何兼容 DDE 的程序共享过程数据。

4）易连接。可从 Internation 驱动程序库中选择 I/O 驱动程序。Internation 的驱动程序支持复杂的冗余连接、通信故障检测和恢复及一个内置的数据观察器等，从而成为工业标准。

在与条码阅读器、磅秤等简单设备交换过程数据时，可使用 DDE。

系统能与 Oracle、Sybase、SQL Server、Ingres、Access 等关系数据库系统共享数据。实时 SQL 接口采用 Microsoft 的 ODBC（Open Database Connectivity）提供系统和关系数据库之间的双向访问。

5）易生长。将系统扩展成多节点网络式 SCADA 应用，就像把一台 PC 机连接到网络中同样容易。所有产品都被设计成既支持单节点应用，又支持网络应用。分布式客户机/服务器体系结构，能即刻在整个企业中共享数据。

可知，无论产品的功能，还是从产品的性能看，FIX、iFIX 确是世界级专业组态软件。

（3）SIMATIC WINCC。是德国西门子公司开发的。是它的 SIMATIC PCS 7 过程控制系统及其它西门子控制系统中的人机界面组件。其特点有：

1）WinCC 提供有丰富的选件（options）和附加件（add-ons）。

2）WinCC 应用范围广泛。它的系统设计，模块化结构，以及灵活的扩展方式，使其不但可做单用户应用，还可做多用户应用。甚至在工业和楼宇技术中，包含有几个服务器和客户机的分布式系统也可用它。

3）WinCC 集生产自动化和过程自动化于一体，实现了相互之间的整合。

4）WinCC 的组态界面完全是国际化的：可在德文、英文、法文、西班牙文和意大利文之间进行切换。亚洲版还支持中文、韩文和日文。可以在项目中设计多种运行时目标语言，即同时可使用几种欧洲和亚洲语言。

5）WinCC 提供了所有最重要的通信驱动，用于连接到 S IMAT IC S5/S7/505 控制器（例如通过 S7 协议集）的通信，以及如 PROFIBUS-DP/FMS、DDE（动态数据交换）和 OPC 等非专用驱动；也能以附加件的形式获得其它通信驱动。

SIMATIC WinCC 在其基本系统内，集成有基于 Microsoft SQL Server 2000 的功能强大、可延展的“Historian”系统，并以跨公司“Historian”服务器的形式用作中央信息交换系统。不同的评估用客户机、开放性接口（开放性数据库接口：ADO，OLEDB，SQL；编程接口：VBScript 和有访问 COM 对象模型和 API 功能的 ANSI-C）以及各种任选件（WinCC/Dat@ Monitor，WinCC/Connectivity Pack，WinCC/IndustrialDataBridge）构成了灵活而高效的 IT 和商务集成的基础，这样就可以与生产和公司管理层软件（MES 和 ERP）相连接。

（4）RSView32。RSView32 是 AB 公司的组态软件。AB 公司是世界最大的 PLC 生产公司之一。它的 RSView32 是基于 Windows 环境（支持 Windows 2000）的工业监控软件。RSView32 同时提供英文版，中文版，法文版，德文版，意大利文版，日文版，葡萄牙文版，韩文版和西班牙文版。利用 RSView32 可以广泛的和不同的 PLC，包括第三方的 PLC，建立通信连接，建立广阔的监控应用。

RSView32 突出特点是：

1）全面支持 ActiveX 技术，使得用户可以在显示画面中任意简单地插入 ActiveX 控件，来丰富应用。

2）开发了 RSView32 的对象模型（Object Model），使得用户可以简单的将 RSView32 和其它的基于组件的应用软件互操作或者集成应用。

3）集成微软的 Visual Basic for Applications（VBA）作为内建的脚本语言编辑器。可以随意定制开发后台应用程序。

4）同时支持 OPC 的服务器和客户端模式。亦即既可以通过 OPC 和硬件通信，又可以向其它软件提供

OPC 的服务。

5）支持附加件结构-AOA。使得用户可以将其它功能模块直接挂接到 RSView32 的核心上去，生成一体的应用。

可利用远程客户扩展 RSView32 的应用：

RSView32 Active Display System 是用于 RSView32 的客户/服务器应用。利用这个系统，可以从远程客户端非常高效实时地监控到现场的设备运行状况-不但可以读取到实时的数据变化，也可以控制现场。

RSView32 WebServer 对于有权用户提供了不限制客户连接数量的，基于网络浏览器（任何支持 HTML、在任何平台下-包括 Linux/Unix 等等的浏览器）的远程监控方案。可以在远程看到现场的画面，参数值，报警。

等等。

2. 国内组态软件

（1）世纪星。世纪星全称为《世纪星通用工业自动化监控组态软件》，是北京世纪佳诺科技有限公司自主产权的软件产品。1999 年投入市场。至今已有近五千套应用，在相关行业使用。这些行业是：电力、石油、化工、冶金、矿山、工业民用水处理、环保污水处理、储备粮库、铁路隧道信号监控、交通信号监控、食品及饮料等。

它由开发系统 CSMAKER 和运行系统 CSVIEWER 两部分组成。CSMAKER 和 CSVIEWER 是各自独立的 Windows 32 位应用程序，均可单独使用；两者又相互依存，在开发系统中设计开发的工程和画面的应用程序，必须在 CSVIEWER 运行环境中才能运行。

开发系统 CSMAKER 是应用程序的集成开发环境。在这个环境中，进行画面设计、数据库定义、动画连接、脚本编写等。

运行环境 CSVIEWER，用于显示在开发系统中建立的画面，并负责数据库与 I/O 服务程序的数据交换。它用实时数据库管理，从工业控制对象采集各种数据，并予以显示，同时进行有关报警、历史数据记录、趋势曲线显示，并可生成历史数据文件。

1）主要特点

（a）32 位 Windows 应用系统，运行于中文 Windows 98/NT/2000 平台，全中文界面，实时多任务、多线程，采样速度更快，系统稳定。

（b）智能型人机接口，可视化 IE 风格界面，系统调色板提供 32 位真彩色，渐进色及屏幕抓取点位图等功能，丰富的图库及图形控件，视频信号监视，全屏幕编辑功能。

（c）驱动程序采用 COM 组件技术，采用 OLE 自动化技术把《世纪星组态软件》与驱动程序整合在一起，配置方便的设备安装向导，使用户能方便地连接各种硬件设备。

（d）支持国内外常用的硬件设备，所有驱动程序免费提供。

（e）提供多串行接口、Modem 拨号、无线电台、电力载波等多种解决方案；支持各种现场总线（Profi-Bus、LonWorks、CanBus 等）；支持 DDE、OPC、ODBC、Web、TCP/IP 局域网等接口规范。

（f）提供双机热备、多级安全保障方式。

（g）自定义函数、直接 I/O 读取函数、计时器、定时器函数、计数器、积算函数、简单 PID 函数、便捷的配方管理、渐进色填充、立体管道、成组定义变量及变量过滤、支持数组、从屏幕抓取点位图等创新功能。

（h）提供全新组态报表，在工控软件界首次引入组态报表概念，组态报表使用方便（类似 Excel），功能强大，不但可以得到实时报表和历史报表（班报、日报、月报、年报等），同时，在报表中可任意插入柱状图、圆饼图、折线图、散点图等，并能随意打印。

（i）内置定时报表打印、画面打印、曲线打印、报警记录打印等打印功能。

（j）温控曲线、XY 曲线、棒图曲线、窗口控制等控件。

为了进一步理解这个软件，并便于使用，它的几个与使用有关的问题再分述如下：

2）软件基本结构。它的基本结构是以数据库（DataBase）为核心，向上表现为人机界面（HMI，包括画面制作、报表、趋势曲线、报警）及其它应用（如网络、ODBC 等功能），向下表现为与其它应用程序的动态数据交换（DDE）及与现场设备的驱动程序（I/O Driver）。其基本结构如图 7-73 所示。

3）变量数据库。它的变量数据库又称变量字典，是《世纪星组态软件》的核心，是变量的集合。数据库中的每一变量包括变量名、数据类型、变量取值范围、当前值、连接的设备（对 I/O 类型变量）等。运行时程序维护一个实时数据库，各个功能模块随机访问数据库，数据库管理系统保证数据的更新。

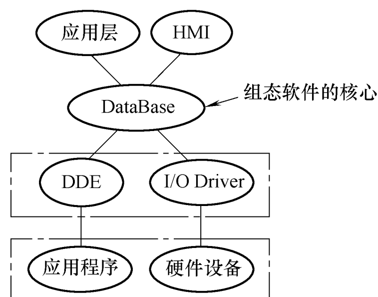


图 7-73　世纪星组态软件基本结构

数据库的变量类型有：系统变量、内存变量、I/O 变量、特殊变量等四种，其中除特殊变量外都有离散、整数、实数、信息等四种类型，系统变量除含以上四种类型外，还有系统报警组变量，特殊变量包括历史趋势曲线变量和报警窗口变量。

此外，变量还有“域”的概念。变量的域反映变量的属性，在变量定义时，设置变量的属性即是设置变量的域值。对历史趋势曲线变量及报警窗口变量来说，除在定义变量时设置一些基本属性（如变量名、变量描述等），其它属性在动画连接时设置。

对变量数据库的操作包括，新建、修改、删除、列表方式选择、退出等。系统变量都由 $ 字符开头，用户不能修改或删除；用户定义的变量的 ID 号从 101 开始向下排列，定义一个变量后（已经保存），除变量类型不能修改外，变量的其它属性都能修改。

变量数据库的维护大部分由系统自动完成，用户只需运行“更新变量计数”后再运行“删除未用变量”即可，当然，在开始作以上工作时必须关闭所有画面。

4）人机界面开发工具。它提供了可视化 IE 风格人机界面开发工具。用它可设计监控和数据采集系统所必须的图形界面。

这些工具为：1670 万种颜色的调色板；易于操作的绘图工具，如直线、垂直及水平直线、折线、椭圆、矩形、圆角矩形、多边形；还有位图、历史趋势曲线、实时趋势曲线、报警窗口、文本、按钮。庞大的图形控件库，为用户提供了上百种图形控件，其中专业图形控件多达十多种，每种图形控件中有多个图形控件单元。这些图形控件单元均可实现无级缩放，为了不改变一些专用设备如时钟、仪表、管道、阀门等的图形形状，对这些图形控件单元进行了按比例放大或缩小处理，犹如一个“图形的世界”，极大的加快应用系统的构造。

图形控件中每个单元都具有专用参数输入对话框，在这些对话框中，开发人员输入少量参数，本系统自动处理这些参数，生成图形控件单元的属性连接和动画连接。

为了满足不同用户的需求，图库可以进行自由扩充，设计者可将自己设计的图形存入自定义图库中或新建图库，也可将不需要的图形删除。

还有点位图（Bmp）处理功能，具有硬盘直接装载、粘贴、复制、显示原始大小点位图等功能。此外，还特别提供从屏幕直接抓取点位图和透明处理功能。点位图透明处理是为了在本系统加入不规则的点位图图形。从屏幕上任意抓取点位图，为用户提供了丰富便捷的应用设计工具，帮助用户进行快速图形编辑，大大的提高运行效率。如能结合扫描仪，在开发效率上更高。

5）动画连接。动画连接就是建立画面的图素与数据库变量的对应关系。这样，工业现场的数据，比如温度、液面高度等，变化时，通过 I/O 接口，将引起实时数据库中变量的变化，如果定义了一个画面图素，比如指针与这个变量相关，将会看到指针将与数据变化同步变化。

动画连接的引入把程序员从重复的图形编程中解放出来，为程序员提供了标准的工业控制图形界面。同时，图形对象与变量之间有丰富的连接类型，给程序员设计图形界面提供了极大的方便。

图形对象可以按动画连接要求改变颜色、尺寸、位置、填充百分比、旋转、闪烁等。一个图形对象还可以同时定义多个连接。把这些动画连接组合起来，控制界面将呈现出更好的动画效果。

6）趋势曲线。趋势曲线有实时趋势曲线和历史趋势曲线两种。趋势曲线的外形类似于坐标纸，X 轴代表时间，Y 轴代表变量值。主要用于观察变量的变化趋势。同一个趋势曲线中最多可同时显示四个变量的变化情况，而一个画面中可定义数量不限的趋势曲线。

实时趋势曲线用于实时显示数据的变化情况，曲线会自动卷动，便于观察变量的变化趋势，不需专门定义实时趋势曲线变量；每个实时趋势曲线可反应四个由变量及运算符组成的表达式的变化趋势，而一个画面可定义数量不限的趋势曲线。

历史趋势曲线用于查看历史数据，曲线一般不自动卷动，与功能按钮一起工作，利用历史趋势曲线变量的域或使用与历史趋势曲线有关的函数可以完成对历史趋势曲线的控制，如翻页、起动/停止记录、打印曲线等功能；使用前必须先定义历史趋势曲线变量，每个历史趋势曲线可反映八个已选择记录属性的变量（而不能是表达式）的历史数据变化趋势。

7）报警处理。报警处理是为了检测出非正常状态的发生，且将报警信息登录于有关文件或数据库中。即使在多种画面中，也可以直接通知操作员系统中正在发生的报警。必要时辅以声光报警。

为了方便报警信息的管理，它引入报警优先级达 999 级。由于报警信息的重要性，操作者都希望提供多种报警方式。

（a）实时报警。实时报警包含了声音和视觉报警。声光报警都是通过函数来实现的。视觉报警是通过在屏幕上的一具不停闪烁的指示灯，以及弹出报警信息框实现。在报警信息框中有，本次报警的测点号、报警原因（如上／下限报警）和报警等信息。当没有报警时，报警会自动解除，但是系统会记录报警的起始时间等信息，以备以后查询。当发生报警时操作员可以手动关闭报警，或是完全禁止报警的发生。

（b）历史报警。对报警必须要有一个完整的记录，以便以后查询。要记录的报警事件的参数有以下几种：产生报警的对象、报警发生的时间、报警时报警变量的大小、报警事件的性质等。还提供了报警记录数据的可视化查询工具，用户只要通过简单的鼠标点击就可实现复杂的数据查询。

对模拟量规定了七种报警：低、低低、高、高高、小偏差、大偏差、变化率；离散量规定了开和关两种报警。

8）脚本语言。除了在定义动画连接时支持连接表达，它提供脚本语言，允许用户定义命令语言来驱动应用程序，增强了应用程序的灵活性。此语言既和 C 语言一样简练、灵活，同时又具有 Basic 语言易学易用的特点。

它提供较丰富的内部函数，有：

（a）数学函数：三角函数、对数和指数函数。

（b）字符串函数：对字符串进行分析、查找、替换、截取以及字符串和数值之间的转换。

（c）控件函数：操作历史曲线、报警窗口、画面的函数、打印函数等。

（d）系统函数：文件操作取系统信息以及控制其它应用程序的函数。

（e）用户自定义函数：用户可自定义函数，增强应用系统的整体功能，达到更精确控制应用系统的目的。

脚本语言还有完备的词法语法查错功能。可以指定执行脚本的条件，根据执行条件的不同，可以分为：

（a）应用程序命令语言：可以在程序起动时执行、关闭时或者在程序运行期间定时。如果希望定时执行，还需要指定时间间隔。

（b）热键命令语言：被链接到设计者指定的热键上，软件运行期间，操作者随时按下热键都可以起动这段命令语言程序。

（c）事件命令语言：规定在事件发生、存在、和消失时分别执行的程序。离散变量名或表达式都可以作为事件。

（d）数据改变命令语言：只链接到变量或变量的域。在变量或变量的域的值变化到超出变量字典中所定义的变化灵敏度时，它们就被执行一次。

9）工程配方管理。在制造领域，工程配方是用来描述生产一件产品所用的不同配料之间的比例关系。

是生产过程中一些变量对应的参数设定值的集合。它支持对工程配方的管理。可在工程配方模板文件中定义和存储，每一个工程配方模板文件以扩展名为 csv 的文件格式存储，一个工程配方模板文件是通过工程配方定义模板产生的。

工程配方模板文件中的工程配方定义模板完成后，在运行时可以通过工程配方函数进行各种工程配方的调入，修改、存储等。工程配方分配的功能由工程配方函数来完成，通过工程配方分配将指定工程配方（如配方 M）传递到相应的变量中。

10）I/O 设备驱动。设备驱动采用 OLE 自动化（即 COM 组件）接口技术。COM（Component Object Model 组件对象模型）是一种以组件为发布单元的对象模型，这种模型使各软件组件可以用一种统一的方式进行交互。COM 既提供了组件之间进行交互的规范，也提供了实现交互的环境，因为组件对象之间交互的规范不依赖于任何特定的语言，所以 COM 也可以是不同语言协作的一种标准。

这里的 COM 组件驱动设备分为三大类：

（a）串行接口通信设备：串行通信方式是《世纪星组态软件》与 I/O 设备之间最常用的一种数据交换方式，I/O 设备通过 RS-232 串行通信电缆连接到使用《世纪星组态软件》的计算机串行接口。《世纪星组态软件》最多可与 32 个串行接口设备相连。

（b）板卡驱动设备：板卡驱动设备是插入计算机总线扩展槽中的 I/O 设备。

（c）现场总线：如 ProfiBus、CAN、LonWorks、FF 等现场总线。

11）DDE 动态数据交换设备。在 DDE 数据交换时，提供数据的应用程序为服务应用程序，接收数据的应用程序为客户应用程序。《世纪星组态软件》通过服务程序名、话题名、项目名标识 I/O Sever 中的数据变量，I/O Sever 通过上述三个会话参数从客户服务程序获取数据。《世纪星组态软件》应用程序名为 CS-VIEWER，话题名为 TAGNAME，项目名为已定义 DDE 的 I/O 变量名。

《世纪星组态软件》支持通过 DDE 与标准 Windows 应用程序进行动态双向的数据交换，例如，《世纪星组态软件》与 Excel 用 DDE 进行数据交换时，设备对象名称可输入任意字符如“Excel 程序”，服务程序名用“Excel”；话题名用“Sheet1”表示与 Excel 中的 Sheet1 进行数据交换；完成设备安装后，在《世纪星组态软件》中定义一个 I/O 变量，设备对象名选择“Excel 程序”及在项目名中输入“RxCy”（x 为行号，y 为列号），保存后，即已把变量与 Excel 中的 RxCy 建立了一种联系。《世纪星组态软件》与 Excel 都运行后，当变量或 Excel 中 RxCy 的值有一个发生变化，另一个也同步地发生改变。

12）网络功能。不同计算机上《世纪星组态软件》中的变量之间传递数据是通过网络功能实现，《世纪星组态软件》所支持的网络协议是 TCP/IP。要实现网络通信，必须有先像定义其它 I/O 变量一样，先在设备安装向导中选中“网络”，按“下一步”，在弹出对话框的“设备对象名”中输入一个名称以标识该网络设备对象，在“节点机器名”中输入要连接的计算机名，按“下一步”就会弹出一个由您确认的对话框，单击“完成”即完成了一次网络设备的安装。

在变量定义时，您定义一 I/O 变量，选择设备对象名为您已经定义的网络设备，在远程变量名中输入对方计算机中《世纪星组态软件》的变量名。在系统中，您就能象用其它变量一样用该变量（包括显示、计算、报警等功能）。

13）双机热备。双机热备是，主机通过连好的网络（至少包括一台主机，一台从机，一台采集站），监测采集站的工作，从机始终保持监视状态，监视主机的工作情况。一旦发现主机异常，从机将在很短的时间内代替主机，进行实时监测并保存历史数据；一旦主机重新起动，而从机检测到主机的存在，则会自动将主机丢失的历史数据拷贝给主机，同时从机将重新处于监视状态。这样即使是发生了事故，系统也能保存一个相对完整的数据库。

双机热备的实现，防止了因现场，以及硬件等各种原因导致数据丢失的情况，增加了系统的可靠性，便于系统维护，双机热备主要功能是实时数据的热备和历史数据的热备。

实时数据的热备：主机与从机的《世纪星组态软件》工程文件完全一致，从机获取实时数据是通过网络从主机获取。正常工作时，从机通过网络从主机获取实时数据。从机与主机之间采取请求与应答的方式

通信。从机每间隔一定的时间，向主机发出请求，主机予以应答。如果主机没有作出应答，说明主机出现故障。从机即切断所有连接主机的网络数据传输，改由从下位设备直接获取数据。从机就是用这种方式实现了实时数据的热备。

在实时数据热备中，各台计算机应保持时钟一致，这就涉及时钟服务器的概念，一般的设置是将主机定为时钟服务器，主机采取广播的方式以一定的时间间隔向各台机器发送校时帧，保持网络的始终统一。而当主机失效时，从机将代替主机成为网络的时钟服务器。

历史数据的热备：双机热备时主机、从机分别保存历史数据，当主机失效时，从机代替主机进行数据采集，同时继续保存从下位机上传的数据；当主机重新恢复，从机监测到主机的存在，首先从机停止从下位机采集数据，并通过网络数据流从主机获取数据；然后从机通过比较主机与从机保存的历史数据文件，向主机的数据库拷贝其丢失的数据，从而实现了历史数据库之间的热备。

从以上介绍，可知组态软件的基本功能与基本特性“世纪星”是都具备的。下一小节，还将以它为例，介绍怎样用组态软件编写应用程序。

（2）组态王。是北京亚控科技发展有限公司自主知识产权组态软件。是国内较早出现的组态软件产品之一。已有九千多个现场（钢铁，化工，电力，国属粮库，邮电通信，环保，水处理，冶金等各行业）应用实例。支持 1500 多种硬件设备（包括 PLC、总线设备、板卡、变频器及仪表）。组态王基于网络的概念，是一个工业级软件平台。

组态王版本较多，如通用版、专用版、网络版、嵌入版，等等。变化也较快，如今为 6.53。如组态王 6.0，据介绍，具有如下十大特点：

1）工程管理。一个系统开发人员可能保存有很多个组态王工程，对于这些工程的集中管理以及新开发工程中的工程备份等都是比较繁琐的事情。组态王工程管理器的主要作用就是为用户集中管理本机上的所有组态王工程。工程管理器的主要功能包括：新建、删除工程，对工程重命名，搜索指定路径下的所有组态王工程，修改工程属性，工程的备份、恢复，数据词典的导入导出，切换到组态王开发或运行环境等。另外，组态王 6.0 开发系统提供工程加密，画面和命令语言导入、导出功能。

2）画面制作系统

（a）支持无限色和过渡色　组态王 6.0 调色板支持无限色，支持二十四种过渡色效果，组态王的任一种绘图工具都可以使用无限色，大部分图形都支持过渡色效果，巧妙地利用无限色和过渡色效果，可以轻松构造逼真、美观的画面。

（b）图库　使用图库具有很多好处：降低了工程人员设计界面的难度，缩短开发周期；用图库开发的软件将具有统一的外观，方便工程人员学习和掌握；利用图库的开放性，工程人员可以生成自己的图库元素，“一次构造，随处使用”，节省了工程人员投资。6.0 图库全新改版，提供具有属性定义向导的图库精灵，用户只需稍做调整即能制作具有个性化的图形。

（c）按钮和图形　组态王 6.0 支持按钮的多种形状和多种效果，并且支持位图按钮，用户可以构造漂亮的按钮。另外，组态王 6.0 支持多种图形格式，如 Gif、Jpg、Bmp 等，用户可以充分利用已有的资源，轻松构造自己功能强大且美观的应用系统。

（d）可视化动画连接向导通过可视化图形操作，直接完成移动、旋转的动画连接定义。

3）报警和事件系统。组态王 6.0 报警系统全新改版，具有方便、灵活、可靠、易于扩展的特点。组态王分布式报警管理提供多种报警管理功能。包括：基于事件的报警、报警分组管理、报警优先级、报警过滤、新增死区和延时概念等功能，以及通过网络的远程报警管理。组态王还可以记录应用程序事件和操作员操作信息。报警和事件具有多种输出方式：文件、数据库、打印机和报警窗，并且可以利用控件等工具轻松浏览和打印报警数据库的内容。

4）报表系统。组态王 6.0 提供一套全新的、集成的内嵌式报表系统，内部提供丰富的报表函数，用户可创建多样的报表。提供报表工具条，操作简单明了，比如：日报表的组态只需用户选择需要的变量和每个变量的收集间隔时间；提供报表模板，方便用户调入其它的表格。报表能够进行组态，例如有日报表、

月报表、年报表、实时报表的组态，另外，报表打印时可以进行预览和页面设置。

5）控件。组态王 6.01 支持 Windows 标准的 Active X 控件（主要为可视控件），包括 Microsoft 提供的标准 Active X 控件和用户自制的 Active X 控件。Active X 控件的引入在很大程度上方便了用户，用户可以灵活地编制一个符合自身需要的控件，或调用一个已有的标准控件，来完成一项复杂的任务，而无须在组态王中做大量的复杂的工作。一般的 Active X 控件都具有属性、方法、事件，用户通过控件的这些属性、事件、方法来完成工作。组态王 6.0 版本中新增三个功能强大的控件，即数据表格控件（可将 ODBC 数据源里的大量数据在组态王中进行显示和打印）；历史曲线控件（可动态增删曲线，进行曲线比较，并且数据来源可以是 ODBC 数据源）；PID 调节控件（对过程量进行闭环控制，可实现三种 pid 控制算法：标准型，归一参数型，和近似微分型）。

6）OPC。全面支持 OPC 标准（组态王 6.0 既可以作为 OPC 服务器，也可以作为 OPC 客户端）开发人员可以从任何一个 OPC 服务器直接获取动态数据，并集成到组态王中；同时组态王作为 OPC 服务器，可向其它符合 OPC 规范的厂商的控制系统提供数据。OPC 节省了不同厂商的控制系统相连的工作量和费用。并且组态王提供 SDK 开发包，用户可以自己利用 VC，VB 编制程序，利用组态王的 OPC 接口来访问组态王的变量和变量的域。

7）通信系统

（a）支持远程拨号组态王 6.0 支持与远程设备间通过拨号方式进行通信。组态王的远程拨号与组态王原有驱动程序无缝连接，硬件设备端无需更改程序。利用远程拨号能实时显示现场设备运行状况，随时打印，报警和历史数据自动上传等功能。

（b）开发中进行硬件测试开发系统中有硬件测试界面，在不起动运行系统的情况下，能测试对硬件设备的读写操作，并且 I/O 变量支持时间戳和质量戳，能随时判断数据采集的时间和检查通信质量的好坏。

（c）支持网络 DDE，组态王 6.0 版本支持 win2000 操作系统下的 DDEshare 方式，实现组态王与 excel 和 vb 程序间通过网络进行数据交换。

8）安全系统。组态王 6.0 采用分级和分区保护的双重保护策略。新增用户组和安全区管理，999 个不同级别的权限和 64 个安全区形成双重保护，另外组态王能记录程序运行中操作员的所有操作。

9）网络功能。组态王 6.0 完全基于网络的概念，是一种真正的客户—服务器模式，支持分布式历史数据库和分布式报警系统，组态王的网络结构是一种柔性结构，可以将整个应用程序分配给多个服务器，如指定报警服务器和历史数据记录服务器，这样可以提高项目的整体容量结构并改善系统的性能。

10）冗余系统。组态王 6.0 提供全面的冗余功能，能够有效地减少数据丢失的可能，增加了系统的可靠性，方便了系统维护。组态王提供三重意义上的冗余功能，即双设备冗余、双机冗余和双网络冗余。对于这三种冗余方式，设计者可综合运用，可以同时采取或采取其中的任意一种或两种。采用冗余后，系统运行时将更加稳定、可靠，对各种情况都能应付自如。

组态王 6.01 提供一套开发工具，包括：驱动开发包，图库开发包，SDK 开发包（利用 VC 或 VB 访问组态王的变量和域），DDE 开发包，提供详细操作说明和示例文件，用户无需参加培训即可使用。

后推出的组态王 6.03 又新增加了 9 个功能：I/O 数据采集性能优化；SQL 数据插入的性能优化；加快运行系统初始化速度；支持多个 OPC Server 设备；增加超级 XY 曲线控件；支持 Windows XP 操作系统；新增加密锁序列号获取函数；更多的设备驱动程序；更新更完美的设备驱动帮助。

组态王 6.5 又有新进步。它采用 JAVA 2 核心技术。它是基于浏览器/服务器模式的一种新型的客户/服务器体系结构。它改变原来的 C/S 系统的两层结构为三层结构。客户端以通用的浏览器为基础（IE），服务器端由 Web 服务器及数据库两层结构组成。

使用它，企业的自动化监控将以一个门户网站的形式呈现给使用者，并且不同工作职责的使用者使用各自的授权口令，完成各自的操作。如，现场的操作者可以完成设备的起停，中控室的工程师可以完成工艺参数的整定，办公室的决策者可以实时掌握生产成本、设备利用率及产量等数据。

组态王 6.5 还有很多其它功能，在此就不一一介绍了。此外，组态王还提供一些开发包，可用其进行

有关开发。这些是：

1）驱动开发包。该开发包采用微软标准的 COM 组件技术，用于开发组态王的驱动程序。在创建接口时，可以创建多个互相独立对象，每个对象都可以拥有自己的变量。最后的结果是一个 DLL 文件。接口中的各函数，被组态王的两个应用程序——TouchExplorer. exe 和 TouchVew. exe 调用。

驱动程序只能使用 VC + + 开发。因为它有两个 VC 的头文件：IcomPro. h 和 datatype. h 和一个 demo 项目及一个制作安装文件的项目代码。

2）图库开发包。如果用户需要用到比组态王图库更多的图形，用户可利用这个开发包提供的程序和说明，用 VC 和组态王的图素生成的代码编写图形程序。生成文件后，再加入到组态王图库中。

3）SDK For 组态王开发包。组态王 6. 01 具有 OPC 服务器的功能，但对于用户应用程序不支持 OPC 的情况来说，完全访问组态王中的数据比较困难。为了使用户能够更方便快捷的访问组态王的数据，可利用这个开发包。

它是一个开放的应用程序接口。该接口以动态连接库（. dll）的形式提供给用户。用户可以用 VB 或 VC 等开发独立的应用程序，来直接访问组态王运行系统中实时数据库中的变量或变量的域值。该独立应用程序可以和组态王 6. 0 实现无缝整合，接口中提供了丰富的函数。

4）DDE 开发包。DDE 是 Microsoft 公司设计的一个完整通信协议，它能使两个或多个应用程序之间相互传送数据和指令。当一个应用程序如“组态王”，想从另一个应用程序，如松下 FP3 的“Server”得到数据。在它们之间则必须建立 client-server 关系，也就是建立 DDE 连接，提供数据的一方称为 server，接收数据的一方称为 client。Client 应用程序通过规定“服务程序名”，“话题名”，“项目名”，才可从 server 中获得某一项的数据。比如将“组态王”作为服务程序，EXCEL 作为客户程序，EXCEL 要从“组态王”中取得数据，则可在 EXCEL 的某单元格中规定“ = View | Tagname！DDE1”，其中 View 是“组态王”的服务应用程序名，Tagname 是“标题名”，DDE1 是某变量的项目名，则当“组态王”中该变量变化时，EXCEL 中的单元格会有相应的变化。

（3）ForceControl（力控）。是大庆三维公司的开发的组态软件。大约在 1993 年左右，力控就有了第一个版本。但直至 Windows95 版本的力控诞生之前，他主要被用于公司内部的一些工程项目。32 位 Windows 下的 1. 0 版的力控，其特征之一是，其基于分布式实时数据库的三层结构，而且其实时数据库结构为可组态的“活结构”。但 1. 0 版的力控存在明显的不足，如：I/O 驱较少，界面和产品包装不够美观等。在 1999 ~ 2000 年期间，力控得到了长足的发展，最新推出的 2. 0 版在功能的丰富性、易用性、开放性和 I/O 驱动数量，都得到了很大的提高。

力控 2. 0 是一个集成式的软件包，其中所有组件都可以独立分布式地运行，通过网络服务程序与其它组件交换数据。力控 2. 0 可以运行于 Pentium 133 以上的计算机（16M 以上内存、1G 以上硬盘）系统中。可以实现操作站的双机冗余热备用。

力控 2. 0 包括以下几个主要部分：

1）Draw，功能强大的人机界面组态工具。Draw 是集成的开发环境，它使用面向对象的图形，创建动画式显示窗口。这些窗口数据、图形显示内容可以来自过程 I/O 或 Microsoft Windows 第三方应用程序。

2）View，高可靠、快速的图形界面运行系统。View 用来运行由 Draw 创建的图形窗口，支持的画面数量不受限制，数据刷新速度快于 5ms。

3）DB，先进的分布式实时数据库。DB 是整个应用系统的核心，构建分布式应用系统的基础。它负责整个力控应用系统的实时数据处理、历史数据存储、统计数据处理、报警信息处理、数据服务请求处理。完成与过程的双向数据通信。DB 与 Draw 构成服务器/客户计算模式。各个网络节点上的 DB 通过网络服务程序可以构建成复杂的分布式网络应用系统，单机数据处理能力超过 1 万点，历史数据可以保存 10 年以上；网络处理能力可超过 10 万点。

4）NetClient 和 NetServer，高性能的网络通信服务程序。NetServer 和 NetClient 内部采用 TCP/IP 通信协议，它保证用户可以充分利用 Intranet/Internet 的网络资源，保证数据刷新速度快于 5ms，网络数据处理能

力超过 10 万点。

5）I/O Server，即 I/O 驱动程序。I/O Server 完成与各种检测、控制设备的通信，负责从过程 I/O 设备读取实时数据，同时将来自图形界面和实时数据库的控制命令写入 I/O 设备。DB 与 I/O Server 构成服务器/客户计算模式。I/O Server 由很多单体程序构成，每个单体程序能够完成特定设备的通信功能，目前力控 2.0 的 I/O Server 家族拥有众多成员，支持大多数主流控制设备生产商提供的硬件。此外，还有 DDE 和 OPC Client 两个通用的标准 I/O 驱动程序，用来和支持 DDE 标准和 OPC 标准的 I/O 设备通信。

另外，力控 2.0 中也包含其它可选程序组件：

1）策略编辑生成及运行程序 StrategyBuilder，新一代基于 PC-Based 和嵌入式系统的自动化控制软件，符合 IEC1131-3 标准，可提供比 PLC 更为强大、更为灵活的功能。

2）力控 Web Server，运行在 Web 服务器上的应用软件。可为世界各地的远程用户在台式机或便携机上用标准浏览器实时监控现场生产过程。

3）TelClient/TelServer，使用简便的远程拨号通信程序。在任何地方，只要能拨打电话，就可以使用本组件实现对远程现场生产过程的的实时监控，惟一需要的是 Modem 和电话线。

4）SCOMClient/SCOMServer，低成本的串行通信程序。两台计算机之间，使用 RS232C/422/485 接口，可实现一对一（1:1 方式）的通信；如果使用 RS485 总线，还可实现一对多台计算机（1:N 方式）的通信。

通用数据库接口组件用来完成组态软件的实时数据库与通用数据库（如 Oracle、Sybase、Foxpro、DB2、Infomix、SQL Server 等）的互联，实现双向数据交换，通用数据库既可以读取实时数据，也可以读取历史数据；实时数据库也可以从通用数据库实时地读入数据。通用数据库接口（ODBC 接口）组态环境用于指定要交换的通用数据库的数据库结构、字段名称及属性、时间区段、采样周期、字段与实时数据库数据的对应关系等。

（4）昆仑通态。昆仑通态 MCGS5.5 是北京昆仑通态公司开发的具有自主产权的组态软件。有通用版、网络版、嵌入版。

如 MCGS5.5 通用版，使用 Windows XP 风格的界面，外观简洁、大方；MCGS5.5 组态软件不分开发版与运行版，两者合二为一，可实现在线组态功能。

此外，还有如下特点：

1）MCGS5.5 支持用户实现个性化风格布局界面。

2）MCGS5.5 具有模板功能，用户可以方便的应用或定制功能模板，大大的加快开发速度和减少重复工作量。

3）大数据存盘功能，包括：支持通过 SQ · Server、Orac · e 等大型数据库存盘，支持每日一个的小文件存盘。

4）支持旋转和渐进色功能，组态和运行均可以实现任意角度旋转和改变控件的颜色。

5）支持透明色功能，巧妙地利用透明色效果，可以轻松构造美丽画面。

6）支持 GIF 动画功能，使用 GIF 动画使您的组态画面更加生动、明了。

7）支持自定义控件属性的功能，用户可根据自己需要定制构件的各种属性。

8）支持日志功能，支持三种日志：系统日志、操作日志和用户自定义日志。

9）强大的网络功能，支持 TCP/IP、MODEM、485/422/232 等多种网络数据传输方案。

10）支持多种曲线功能，支持历史曲线、实时曲线、计划曲线，以及相对曲线和条件曲线等多种工控曲线。

11）强大的报表功能，工程人员可以制作实时报表、历史报表和自由表格，同时还可以利用丰富的报表函数，实现各种复杂运算、数据转换、统计分析、报表打印等。

12）MCGS5.5 支持多语言版本，用户可以根据需要选择中文版或英文版。

13）支持视频采集功能：支持多种图象数据采集功能，方便用户直观的观察现场的情况。

14）支持多用户协同开发：方便多个开发人员共同完成一个项目，大大提高了项目的开发速度，缩短

了开发周期。

15）众多的设备驱动：在 MCGS 5.5 中，将支持多达 700 多种的设备驱动。

16）开放的设备驱动接口，功能构件接口和 ActiveX 动画接口，使 MCGS 软件的功能可以无限扩充。

等等。

此外，国内有不少单位，如一些高校、研究所、公司，甚至一些个人正在积极地进行组态软件产品的开发。国产化的组态软件具有较强的价格竞争优势，并适合中国国情，因此已在中国的组态软件市场上占越来越多的份额。

7.4.3　组态软件编程

1. 编程准备

要选好组态软件的品牌和版本。有的是单机版，有的为网络版。还有可选组件。要根据需要选用。

要选用合适的点数。点数与系统的复杂程度有关。要监控及采集的数据多，用点数也多。而点数多，售价也贵。最贵的为无限点，使用的点数不受限制。

要安装好软件。安装前，要按组态软件要求，配置好操作系统。然后，按要求安装组态软件。

要做好解密。作为软件商品，所有组态软件都是加密的。没有解密，有的打不开，或打开了，但只能运行一两个小时。解密有两个方法：在计算机的并口上，插入厂家提供的解密“狗”（芯片），用硬件解密；用厂家提供的授权软盘，运行授权程序，把软盘上的“权”转移到计算机的硬盘上，用软件解密。有的干脆两者都用。

2. 编写程序

以下以使用“世纪星”为例，看如何用组态软件编程。一般有如下步骤：

（1）建立工程。安装完“世纪星”软件后，在桌面上有两个“世纪星”图标。用鼠标双击“世纪星开发系统”图标或从开始菜单起动世纪星开发系统 CSMAKER，则进入开发环境。出现如图 7-74 所示的“世纪星”窗口。

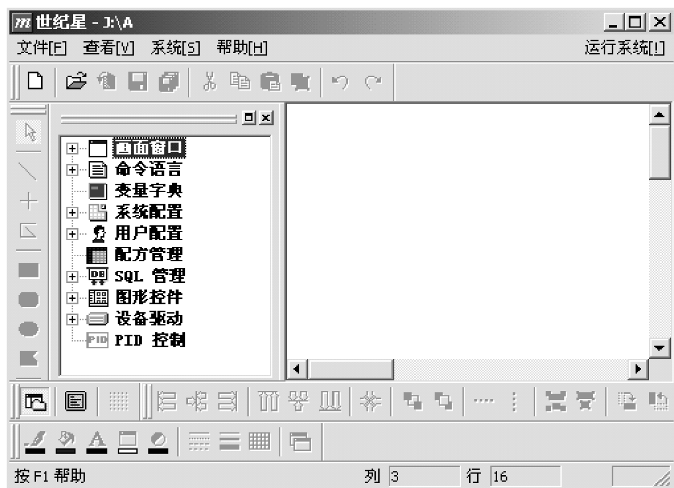


图 7-74　“世纪星”窗口

在“世纪星”窗口上，用鼠标左键击“文件”菜单，再在其上击“工程项目管理器”子项。将出现如图 7-75 所示的“工程项目管理器”窗口。

此窗口可做 5 个选择。新建：用于新建工程；打开：用于打开现有工程，如图 7-75 所



图 7-75 工程项目管理器

示，如选择演示工程 1，再击打开，则弹出演示工程 1 窗口；还有连接、修改、删除，可对工程项目做相应操作。

提示：只当有关闭所有画面时，“文件”菜单中才出现“工程项目管理器”菜单项。

如图 7-75 所示，单击“新建”图标，则弹出“新建工程项目”窗口，如图 7-76 所示。



图 7-76 新建工程项目窗口

在该窗口可键入工程项目名称，并选择工程项目存储路径以及显示分辨率。必要时，还可在工程项目描述中，写入工程项目的有关说明。

当作好以上选定后，单击“确定”，则退出此窗口，回到工程项目管理器画面。建立了工程，设置了工程项目名，用户应用程序的所有信息将被保存在该工程项目目录中。不同的工程项目应设置不同的工程项目名，可避免混淆和数据丢失。

（2）设计画面。画面是监控程序的界面，没有它无法显示数据，也无法进行任何操作。用什么画面和用多少画面按系统监控及数据采集的需要而定。

1）建立画面。在“世纪星”窗口上，用鼠标左键单击“文件→新画面”菜单选项或使用热键 Ctrl + N后，则弹出“新画面”对话



图 7-77 新画面对话框

框，在对话框中可定义画面的名称、背景颜色、风格和画面创建时所处的位置。创建新画面的对话框如图 7-77 所示。

2）设计画面。在画面上，可用世纪星提供的图素，如直线、矩形（圆角矩形）、椭圆（圆）、点位图、多边形、文本等基本图形对象，或图形对象，如按钮、趋势曲线窗口、报警窗口等，建立画面的图形界面。

一个画面要用什么图素或其它图形对象，可按实际需要选定。如要在画面上加入“按钮”，可选择“按钮”菜单命令，此时鼠标光标变为十字形，将鼠标光标置于一个起始位置。按下鼠标左键并拖动鼠标，此时屏幕上出现一个随鼠标移动而变化的矩形框，此矩形框表示所绘制按钮的大小。移动鼠标到新的位置，然后松开左键，即完成按钮的绘制。绘制的按钮如图 7-78a 所示：



图 7-78 按钮设计

如要改变按钮上的文字或文字显示颜色，可选该按钮。然后用鼠标右键击之，将弹出一个菜单。在其中再选“改变文本”项，即弹出“按钮文本”对话框，如图 7-78b 所示。这时，可输入新的按钮名称。如图，这时如击“确定”，则显示的按钮文本将是“起动”，而不再是“文本”。

（3）选择驱动。世纪星采用设备驱动向导，进行 I/O 设备配置。当选择“文件/驱动设备管理”或在浏览器中双击“驱动设备管理”，则弹出驱动设备管理对话框，如图 7-79 所示。

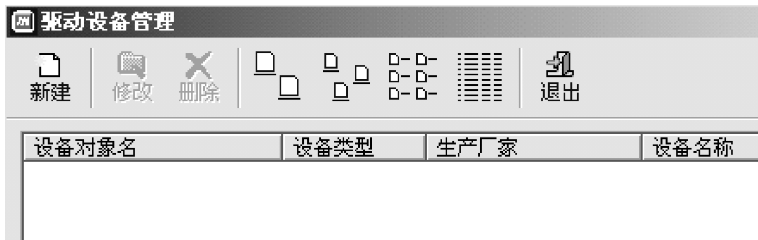


图 7-79 设备管理对话框

接着，在“驱动设备管理”中，选择“新建”项，或在浏览器中双击“设备安装向导”则弹出设备安装向导对话框，如图 7-80 所示：

这时，从树形设备列表区中，可选择 PLC、智能仪表、智能模块、变频器等节点中的一个。然后选择要配置串口设备的生产厂家、设备对象名称。

进而，单击“下一步”按钮，则弹出设备安装向导对话框，如图 7-81 所示：

图中有如下各项，可选填。

1）设备对象名称。要安装串行接口通信设备的名称。

2）通信端口。串行接口设备与计算机相连的串行接口号，该下拉式串行接口列表框共有 32 个串行接口号可供用户选择。

3）设备地址。串行接口通信设备的设备地址。OMRON 可在 0 ~ 31 间选，但应与 PLC 的地址一致。



图 7-80　设备安装向导对话框 1



图 7-81　设备安装向导对话框 2

4）尝试恢复间隔。在世纪星运行期间，如果有一台设备如 PLC1 发生故障，则世纪星能够自动诊断并停止采集与该设备相关的数据，但会每隔一段时间尝试恢复与该设备的通信，如图 7-81 所示，尝试时间间隔为 3s。

5）最长恢复时间。若世纪星在一段时间之内一直不能恢复与 PLC1 的通信，则不再尝试恢复与 PLC1 通信，这一时间就是指最长恢复时间。

6）使用动态优化。世纪星对全部通信过程采取动态管理的办法，只有在数据上位机需要时才被采集。

继续单击“下一步”按钮，则弹出设备安装向导对话框如图 7-82 所示。

从图 7-82 知，这里已建立设备对象名为“新设备”，其地址为 0，是 OMRON 的 PLC。



图 7-82　设备安装向导对话框 3

在此向导页，显示已配置的串行接口通信设备的设备信息。如果需要修改，单击“上一步”按钮，则可返回上一个对话框，可进行修改。如果不需要修改，单击“完成”按钮，则完成设备安装。这时，在“世纪星”窗口的浏览器的“当前驱动设备”中，将增加新安装的设备。

（4）建立变量库。世纪星变量数据库是一个实时数据库。其中变量类型有：系统变量、内存变量、I/O 变量和特殊变量。

1）系统变量。是系统预先设置的变量，这些变量用户可以直接使用。系统变量又分为系统离散、系统整数、系统实数、系统信息和系统报警组变量。系统变量属性有的为只读，有的可读写。如系统时间是只读变量，由系统自动更新，用户不能改变这些变量的数值；对于具有读写属性的系统变量，用户可以改变变量的数值。

2）内存变量。是用户定义在系统内部的变量。存放计算处理的中间值，以及在系统仿真时模拟 I/O 变量。内存变量又分为内存离散变量、内存整数变量、内存实数变量和内存信息变量四种。

3）I/O 变量。是能与其它应用程序进行数据交换的变量。本系统的 I/O 变量能以多种数据交换协议同外部应用程序进行数据交换，如（DDE）、OPC、网络、串行接口、总线、板卡等。具有读写属性的 I/O 变量数据变化时，系统立即将 I/O 变量的值写到外部应用程序。I/O 变量的值也可以由外部应用程序更新。I/O 变量又分为 I/O 离散变量、I/O 整数变量、I/O 实数变量、I/O 信息变量四种。

4）特殊变量。有报警窗口变量、历史曲线变量两种。主要用于系统报警显示和历史趋势曲线显示。

5）变量的域。反映变量的属性。如实数变量的报警具有“高报警限”、“低报警限”等属性，历史曲线变量具有曲线起始时间、曲线时间长度等属性。在定义变量时，同时需要设置变量的域值。可以用命令语言编制程序来读取或设置变量的域，变量的域具有只读和读写两种类型。变量的域的表示方法为：“变量 . 域”。

6）建立变量。在“世纪星”窗口中，选择“系统→变量数据库”菜单，或选择浏览器中的“变量字典”项，弹出变量数据库管理对话框如图 7-83 所示：



图 7-83　变量数据库管理对话框

这时，如击新建，则弹出如图 7-84 所示的“变量数据库”对话框。

从图 7-84 知，它已建立一个 I/O 整数变量。变量名为“输出通道”。设备对象名“新设备”，即以上建立的 OMRON PLC。寄存器为“IR10”，是 PLC 输出通道地址。

其它变量可按需要建立。所谓组态软件点数，就是指可最多可建立的变量数。

（5）数据连接。如果已建立如图 7-85 所示的简单画面。并建立了如图 7-85 所示加入的“输出通道”I/O 整型变量。看如何建立数据连接？



图 7-84 “变量数据库”对话框

该画面的目的是，用起动按钮去起动工作（这里即使 PLC 的 10.00 点 ON），用停车去停止工作（这里即使 PLC 的 10.00 点 OFF），并用工作指示灯的颜色显示这工作状态。

连接的办法是：

双击起动按钮，则弹出如图 7-86 所示的“动画连接”窗口。

再在其中选“命令语言”项，并用鼠标左键单击。则又弹出如图 7-87 所示的“按钮命令语言连接”窗口。

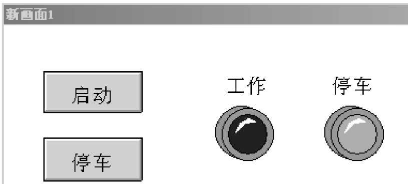


图 7-85 简单画面

可在其中键入：“BitSet（“输出通道”，0，1）;”。再击“语法检查”。如检查通过，击“语言保存”，保存所作的选定。这里“BitSet（“输出通道”，0，1）;”是世纪星的脚本语言的命令语言。其含义是把“输出通道”的第 0 位设为 1。而从变量定义中知道，“输出通道”即为 PLC 的 10 通道。在图 7-87 的函数栏处，击“全部”，即可找道这个函数。



图 7-86 动画连接窗口



图 7-87 按钮命令语言连接

从图看，这个命令语言是在按钮按下时执行的。也可为按钮弹起或按钮按住时执行，可任选。

对停车按钮也可进行连接。方法同此。但它的命令语句为“BitSet（“输出通道”，0，0）;”，即把“输出通道”的第 0 位设为 0。

建立指示灯的连接是：在显示工作指示灯上，击鼠标右键，则弹出如图 7-86 类似的动画连接窗口。选其中的“特殊功能”可见/隐含，再用鼠标左键击之。此时，将弹出如图 7-88所示“可见状态连接”窗口。



图 7-88　可见状态连接

从图知，所列的表达式为真时，目标可见。这里“Bit（输出通道，0）”是世纪星的脚本语言的命令语句。其含义是读“输出通道”的第 0 位。如为 1，则此指示灯显现，否则隐含（画面看不到）。

在显示停车指示灯上，也可作类似设定。只是选表达式为真时，目标隐含。

有了以上画面、I/O 变量及其连接，运行时，如用鼠标左键击“起动按钮”，则 PLC 的 10.00 ON，工作指示灯显现；如用鼠标左键击“停车按钮”时，则 PLC 的 10.00 OFF，停止指示灯显现。如两指示灯位置重叠，则可认为是一个指示灯一样，工作显示蓝灯，停车显示绿灯。

提示：这里选整型量代替开关量，目的是节省“点”。用此方法，一个整型数的“点”可抵 16 个开关量的“点”。而效果完全相同。

（6）脚本设计。世纪星的脚本语言类似于 C 语言。可用它编写命令（脚本）语言程序，以增强应用程序的灵活性。命令语言有五种：

1）应用程序命令语言程序。可以在程序起动时执行、关闭时执行或者在程序运行期间定时执行。如果希望定时执行，还需要指定时间间隔。

2）热键命令语言程序。被链接到设计者指定的热键上，软件运行期间，操作者按下热键即起动这段命令语言程序。

3）事件命令语言程序。规定在事件发生、存在、和消失时分别执行的程序。离散变量名或表达式都可以作为事件。

4）数据改变命令语言。只链接到变量或变量的域。在变量或变量域的值变化到超出变量字典中所定义的变化灵敏度时，它们就执行一次。

5）高速命令语言程序。有关高速采集时用，当起动高速命令语言为 1（真）时，该语言才会执行。

命令语言的句法和 C 语言非常类似，是 C 的一个子集，具有完备的语法查错功能和丰富的运算符、数学函数、字符串函数、控件函数和系统函数。各种命令语言通过“命令语言”对话框编辑输入，在世纪星中被编译执行。

命令语言还提供很多函数，可实现各种功能。如进行画面切换，可使用 Showwindow（“画面编号”）实现。至于什么情况下执行此函数，按要求确定。

（7）系统设定。在“世纪星”窗口中，选择“系统→系统配置→运行系统”菜单，或选择浏览器中的“系统配置→运行系统配置”项，则弹如图 7-89 所示的“运行系统配置”窗口。

从图知，它有 3 个表单项：系统配置，初始窗口及定制菜单。该图显示的为系统配置选定窗口。可按情况填写或选择其中的内容。

选表单“初始窗口”，可在其上选定运行第一个出现的窗口。

选表单“定制菜单”，可在其上选定运行时出现的菜单项。

（8）存储编译。作完了以上这些，基本完成了一个简单的设计。当然，如要报警、显示历史曲线等，还有不少工作要做。

之后，别忘了程序存储。存储后，世纪星会自动编译。到此，组态软件编程即告完成。

3. 运行调试

编程完成后，击如图 7-74 所示“世纪星”窗口右上角上的“运行系统”，则进入所编程序的运行。



图 7-89　“运行系统配置”窗口

如出现问题，再回到开发系统，修改，或重新开发。再运行，再调试，直到达到要求为止。

以上讲的是单机版编程。网络版基本上与它相同，但要作有关网络的配置及处理好服务器、客户机等等问题。

7.5　PLC 与人机界面通信编程

关键词：可编程终端、人机界面、嵌入式人机界面

7.5.1　人机界面概述

用计算机与 PLC 联网、通信，进行监视与控制，一般不便用于工业现场。较好的解决方案是用人机界面（Humen Machine Interface，HMI），也称可编程终端（Programmable Terminal，PT）替代计算机。

人机界面品种、规格很多。有低档的，也有高档的。前者只能显示文字，功能较差，但价格便宜。后者实质上也是计算机，只是它是根据工业现场环境设计制造的。显示屏是液晶的，而且多带有触摸按钮，并与机箱体组装成一个整体。体积很小，工作也很可靠。所以非常适合于现场使用。

较大 PLC 厂家大多数都生产人机界面。其产品可与自身的 PLC 联机使用，也都可以与其它厂家的 PLC 联机使用。此外，还有专门生产人机界面的厂家，它的产品更是可以与任何 PLC（只要有相应的通信接口）联机使用。

图 7-90 所示为 OMRON 公司生产的几种人机界面。其类型比较多。以显示屏的尺寸分，有大、中、小多种。大的可达 194×140mm（640×480 点），小的只有 100×40mm。以色彩分，有单色及彩色的。以显示屏的显示原理分，有液晶、电发光、STN 彩色等。

从操作键来看，有的靠按键进行输入；有的靠触摸按钮输入数据或进行操作；也有带少量功能键，操作靠功能键，而输入数据用触摸按钮。有的还可以实现手持编程器的功能。

从内存来看，其容量各异，少的几十k，多的几百k，以至于几十M。内存大，意味着可显示与处理的画面多，交换的数据多。另外，与PLC交换数据有直接访问方式（Direct Access Method），可以通过软件编程指定要显示或写入的PLC内部器件地址；还有数据内存方法（Data Memory Method），靠设定指定PLC的内部器件区用作交换数据。

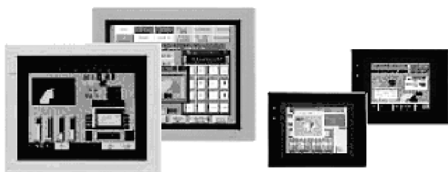


图7-90　可编程终端

从通信协议来看，也是多种多样的：有Host Link方式，还有NT及RS232方式。通信口有RS232，RS422，还可以直接用SYSMAC I/O BUS，即用的是并行的信号。

OMRON公司NT31、NT31C（彩色）可编程终端，为320×240点，57英寸。NT631、NT631C（彩色），为640×480点，113英寸，除了STN彩色，还有TFT彩色。这几个型号都是针对中国市场开发的，自带有简体汉字字库，可显示简体中文。同时，增加了历史曲线及报警数据的记录，而且可以通过并口，实现屏幕打印。此外，它配置有两个RS232口，在与PLC通信的同时，还可以接计算机，可以很方便地实现与计算机通信，或对PLC监控。

此外，OMRON上海公司还开发了微型可编程终端（MPT1、2及5）。它用文字或简单的图形显示，用按键操作。文字可以显示两行，每行若显示中文为10个，数码或英文为20个。由于价格低廉，操作方便，可以在简单的PLC控制系统得以方便地应用。

OMRON还有NS系列产品（N12、NS10、NS12）。都是256彩色的，有效显示面积分别为12.1、10.4及7.7英寸，见图7-91。

该系列产品不仅有串行接口，还有以太网接口、控制网接口，不仅可以通过Host Link网，还可以通过以太网或控制网与PLC通信。具有Windows风格的接口，改善了图形显示效果。还有视频信号显示功能，可以直接显示现场画面。同时，还可以利用它对PLC进行梯形图监控。

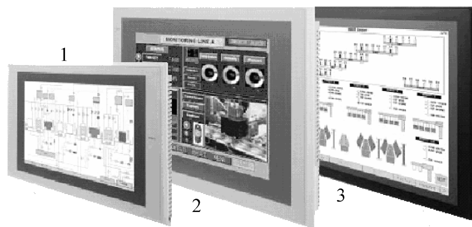


图7-91　高档可编程终端

1—NS7 7.7英寸　2—NS10 10.4英寸

3—NS12 12.1英寸

此外，还有NP系列可编程终端。尺寸：3.8in/5.7in。类型：STN黑白（NP5-MQ）/STN蓝白（NP3）/STN彩色（NP5-SQ）。功能键：6个功能键/3个功能键。CPU：32-Bit RISC，206.4MHz。像素：320×240 pixels。内存：4MB Flash Memory，256KB SRAM。内置接口：通信接口×2（COM1：RS-232C专用，COM2：RS-485/422切换），USB接口×2（USB画面传送，支持U盘）。

沈阳鹭岛资讯科技有限公司人机界面的品牌为“LEODO”，是于2004年开发的嵌入式人机界面，它由32位嵌入式系统和ET组态软件构成。具有信息处理能力强，实时性高、接口丰富、使用方便、安全可靠、低功耗、体积小等特点。由于是国内自己开发的，所以比较实用，而且价格也不高。因而，完全可用作在工业现场使用的PT升级的替代产品。

如图 7-92 所示的为 LEODO（10.4 英寸规格的）外观及其大小尺寸。

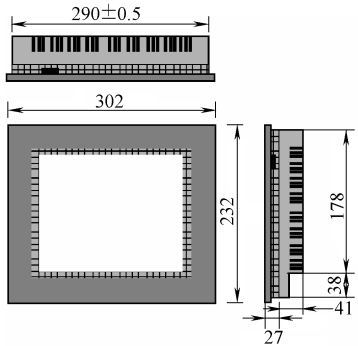


图 7-92　LEODO 外观及其大小尺寸

（1）硬件系统。

LEODO 的硬件核心采用 Intel PXA255 CPU。显示器采用带四线电阻式触摸屏的真彩 LCD。主板内置 64M 内存储器（可扩充）和 64MB 外存储器（也可扩充）。同时，集成有 RS232 接口、RS485 接口、USB 接口、10M 以太网接口、音频接口（接口结构如图 7-93 所示）。

由于 LEODO 的接口如此丰富，除了可用普通的串行接口进行通信，还也可以上互联网，也可外接目前比较流行的 USB 设备。

如图 7-94 所示的为 LEODO 与企业信息网相联的示意图。

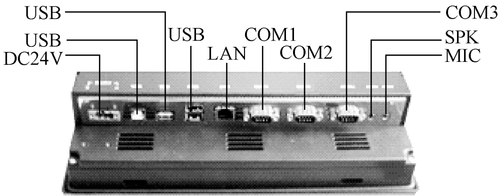


图 7-93　LEODO 接口结构

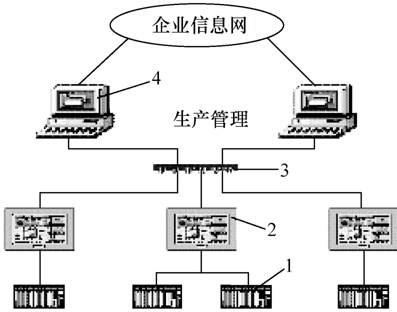


图 7-94　LEODO 与企业信息网相联的示意图

1—PLC　2—LEODO　3—交换机　4—计算机

（2）软件系统。

LEODO 的操作系统为 Windows CE. NET，是微软公司专为嵌入式系统设计的、32 位 Windows 操作系统的简版。与计算机用的 32 位 Windows 相比，虽有所精简，但也是图形界面，也具有多线程，多任务，完全抢占式的特点。

在此平台上，LEODO 还安装了 ET 组态软件的运行版。并提供与运行版相配套的、相当于 PT 的开发软件的开发系统及其模拟运行系统。但是这两者应该安装在计算机上。计算机的操作系统则应是 Windows 2000/Windows NT4. 0/Windows XP。

用户可利用 ET 组态软件开发系统及其模拟运行系统，在计算机上，用普通组态软件的设计方法，进行应用开发，必要时还可以进行模拟运行。运行成功后再下载给 LEODO。

LEODO 已经安装了 ET 组态软件的运行版，再加上下载的用户应用，以及做好 LEODO 与被控设备的连接，经运行，即可实现所预期的人机界面的功能。

（3）与传统触摸屏比较。

传统的 PT 是指目前在工业现场广泛使用的，基于单片机和汇编语言开发的工业触摸屏。这些触摸屏稳定性、可靠性很高，但是由于软、硬件起点低，系统的速度和功能受到了限制。只能用于生产线上简单的图形显示和监控，不能和多种硬件设备相连，不能进行大量的数据存储。因此它已不适应高速发展的控制系统的需要。而 LEODO 的硬件、软件起点都比传统的 PT 要高。

从硬件来看，它的 CPU 档次高，内、外存容量大，而且还可扩展，以至于还可外挂笔记本硬盘；接口丰富。基本上具备了 PC 的硬件平台，而且还可以按定制灵活配置。

从软件来看，由于 LEODO 的操作系统为 Windows CE. NET，所以，不仅可以运行 ET 组态软件，而且，凡是用被其支持的开发系统设计的应用，都可以在 LEODO 运行。这表明 LEODO 建立应用的功能，与传统的 PT 相比都要强得多。

再加上 LEODO 的带有触摸功能的真彩 LCD 的使用，所以，它不仅功能很强，性能很高，而且操作很方便，外观也很漂亮。同时，它所使用的硬件都是工业级的，软件上也有很多安全措施，因而完全能安全、可靠地在工业环境中应用。

以下列的即为 LEODO G 系列的部分产品：

GR-10T-A0　G 系列—RISC　CPU—10. 4 英寸真彩—白色--标准接口
GR-10T-B0　G 系列—RISC　CPU—10. 4 英寸真彩—黑色--标准接口
GR-06T-B0　G 系列—RISC　CPU—6. 4 英寸真彩—黑色--标准接口
GR-12T-B0　G 系列—RISC　CPU—12. 1 英寸真彩—黑色--标准接口
GR-15T-B0　G 系列—RISC　CPU—15　英寸真彩—黑色--标准接口

LEODO 可以单机使用，也可以联网使用。无论怎么使用不外乎是在硬件上做好选定及接线；在软件上用好 ET 组态软件或编好其它应用。

7. 5. 2　人机界面方编程

1. 编程软件

人机界面方编程，多数要用 PT 厂家提供的软件，先在计算机上编程。经编译，再下载给 PT，PT 才能工作。如 OMRON 公司用到软件业集成在 CX-one（CX-Designer）中。但简单的 PT，如 MPT01，没有图形画面，可直接连接 PLC，在与 PLC 联机时编程。编后，即可使用。

编程前要先作设定或选择。因为，这些软件是全方位的，面向很多 PT 机型。但实际用的只能是其中的一种。

PT 编程，应先脱机，用 PT 软件，在计算机上编程。其情况与组态软件编程颇类似。脱机编程，相当于组态软件在开发环境下编程；编好的程序经编译、下载给 PT 后，在 PT 中与 PLC 联机运行，相当于组态软件编译好的程序，在运行环境下运行。

2. 编程步骤

（1）画面设计。图 7-95 所示为一个简单的 PT 画面例子。其功能也是想用按下触摸按钮“起动”使 PLC 的 10. 00 点 ON、工作，按下“停车”使其 OFF、停止工作。并用指示灯的颜色反映 10. 00 是否工作。

从图 7-95 知，它与图 7-85 用组态软件建立的简单画面很相似，功能也相同。这当然这是为了便于理解才这么举例的。

至于这个画面怎么建立，图怎么画成了这样的，细节虽与组态软件不同，但大体类似，也是用种种图形对象画图。具体可参阅有关软件的说明。

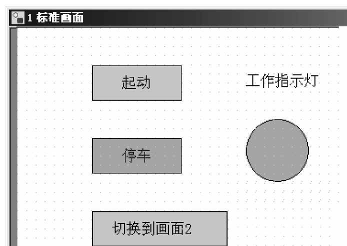


图 7-95　PT 画面

本例还有一个“切换到画面 2”的触摸按钮。按下它，在 PT 上，将显示画面 2。而这时是显示 1 标准画面。其实，组态软件编程，也有画面切换的操作，只是它用命令语句 Showwindow（“画面编号”），实现。而不是这里用的方法（见以下介绍）。

（2）地址对应。图 7-95 中的 PT 画面的功能怎么能实现？这要靠地址对应。这点与组态软件不同：组态软件用变量库中的 I/O 变量及设备驱动设定，而 PT 用地址对应。它用地址对应，使图形对象与 PLC 地址关联。图 7-96 所示为是“触摸开关”的设定操作，目的是使 PLC 的 10.00 位 ON（设置）。



图 7-96　触摸开关设定操作

图 7-97 所示是“标准灯”的灯功能操作，目的用灯的不同颜色反映以 PLC 的 10.00 位 ON/OFF。图中 PLC 地址窗口，是击标准灯窗口的“设定（S）”键后弹出的。此窗口用以直接选择 PLC 地址。选择好，击“确认”键，则关闭此窗口。

至于 10.00 ON/OFF 对应的灯的显示颜色，如图 7-98 所示“标准灯”通用设定。



图 7-97　“标准灯”灯功能操作



图 7-98　“标准灯”通用设定

从图知，12.00 ON 时，显示绿色，12.00 OFF 时显示蓝色。当然，选别的颜色显示也可，在其中作相应选择就是了。

（3）动作关联。除了关联 PLC 地址，有的图像对象还可与有关动作关联。图 7-99 所示为触摸开关设定功能操作时，可能选定的功能。其中“切换画面”、“打印画面”等就是操作动作。触摸开关动作时，与哪个动作关联，在此可选定。

图 7-100 所示为选定触摸开关的功能为切换画面。并设定切换到画面 2。

（4）编译下传。有了以上编程，应存储所编程序，并编译它。也可脱机演示。为了实际进行控制 PT 须与计算机联机，并把已编译的程序下载给 PT。

（5）实际运行。PT 装入程序后可与计算机脱机，并再依配置要求与 PLC 相连接（接

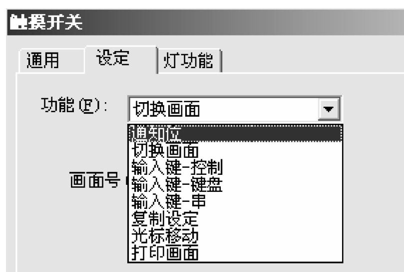


图 7-99　触摸开关设定功能操作



图 7-100　触摸开关的功能为切换到画面 2

线)。PLC 通电，PT 也通电，即可观察 PT 的程序能否达到显示数据、输入数据及存储数据的要求。

如上例，按起动触摸按钮，则 10.00 ON 灯显示蓝色。按停止触摸按钮，则 10.00 OFF 灯显示绿色。按切换到画面 2 触摸按钮，则显示画面 2。

7.5.3　PLC 方编程

PLC 方程序功能有：画面控制；数据准备；数据使用。

数据准备、数据使用，其程序和 PLC 方与计算机通信的程序相同。故这里不再重复介绍。

画面控制是指，PLC 根据控制情况，使 PT 显示应显示的画面，或在 PT 上显示报警或其它操作。这是用 PT 监控与用计算机监控的一个不同点。用计算机监控、画面的切换、由计算机根据读取到的 PLC 数据情况，可自行确定。而 PT 则用设定的 PLC 的某个控制位是否 ON 确定。只要 PLC 的某个控制位 ON，就可使 PT 上与其对应的某个画面显示。

所以，PLC 方在什么时候要求 PT 显示什么画面？怎么利用上述对应关系（当然，不同的 PT，这里的细节可能有所不同）去要求 PT 显示该显示的画面？对于这些问题，PLC 方就要编写相应的控制程序了。

PLC 对画面的控制实质是逻辑关系，如同 PLC 控制一个输出点一样。这样的程序，编起来是不难的。本书第 3 章已作过很多讨论。所以这里不再赘述。

7.6　PLC 与智能装置通信编程

关键词：智能装置、协议宏、协议宏软件 CX-Protocol、CRC16 冗余校验码

智能装置指，智能传感器、智能仪表、智能执行器。多带有串口或相关网络接口。通过这样接口与 PLC 交换数据，连线少，信息量大。用的是数字量，抗干扰能力强。同时，还可增大传送的距离。是 PLC 与传感器、执行器可靠交换数据较好的方法。

与 PLC 通信方法有：用通信指令通信、用协议宏通信，地址映射通信。

PLC 与智能装置通信，编程主要在 PLC 方。智能装置方一般不要编程，但必须弄清它的有关通信协议。智能设备类型多、厂家多、规格多，协议也多，但 PLC 通信编程的要点是一样的，都是根据协议，调用好通信指令。

如果智能装置没有通信口，也可与 PLC 的远程终端连接。这时，它只是 PLC 的远程 I/O，PLC 可直接对其读写数据，无须通信程序。

7.6.1　用通信指令通信

以下用 OMRON 公司“OMP 开发二课”编的此类例子程序，说明 CPM2A/CPM2AH 怎样用通信命令与变频器（3G3MV）通信。

此程序含：

Modbus 协议需要的 CRC16 冗余校验码计算；

用 TXD 命令向变频器（3G3MV）发送控制命令；

用 RXD 命令接受变频器（3G3MV）的响应信息，并保存在 DM 区。

通信算法框图如图 7-101 所示。

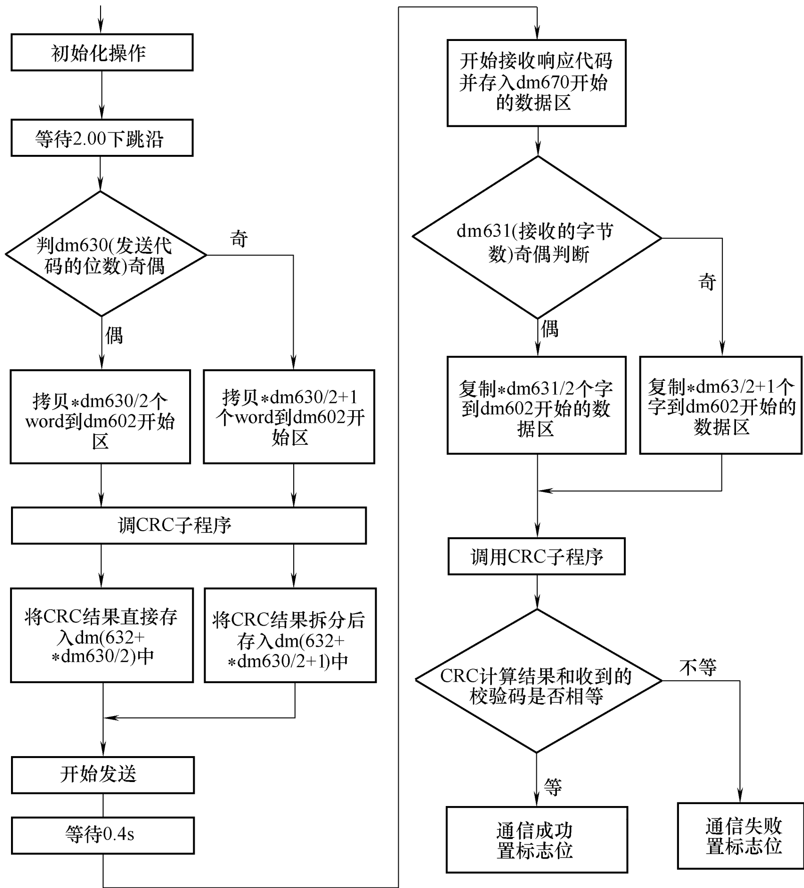


图 7-101　通信算法框图

为了实现的通信及方便客户使用，CPM2A/CPM2AH 通过此程序能正确的向变频器（3G3MV）发送控制命令及接受变频器的返回信息。请注意发送的时序（因 Modbus 协议本身原因，变频器不允许连续接受控制命令）。

通信开始“通信键”（即图 7-101 中的 2.00）按下，如图 7-102 所示。

从图知，这时将使“发送周期”ON，进入发送周期。这时，先按图 7-101 的算法对通信命令（发送报文）进行 CRC 计算，把报文加 CRC 校验码（两个字节）组成发送数据帧（图 7-102 未示出）。当“CRC 结束标志位”ON，微分指令使“CRC 结束微下”ON 一个扫

描周期（如图 7-102 所示）。由它起动数据发送通信（TXD）指令，由 CPM2A 向变频器（3G3MV）发送该数据帧。

经过延时，接收变频器（3G3MV）的响应码，并对该响应码的代码段进行 CRC 校验计算（图 7-102 未示出），用计算的 CRC 代码结果和收到响应码中的 CRC 代码进行比较。根据比较结果进行处理：如果相等，说明通信成功；如果不等，则说明通信失败。

程序使用的资源有：

DM600-DM699

DM630：设定发送的指令的字节数（不包括 CRC 校验码）。

DM631：设定回收的响应代码的字节数（不包括 CRC 校验码）。

DM632-DM641：具体指令设定区。

DM670-DM699：回收响应代码的存储位置

TIM240-TIM249：定时器标志

IR218-IR227：各状态标志

IR2.00：发送触发

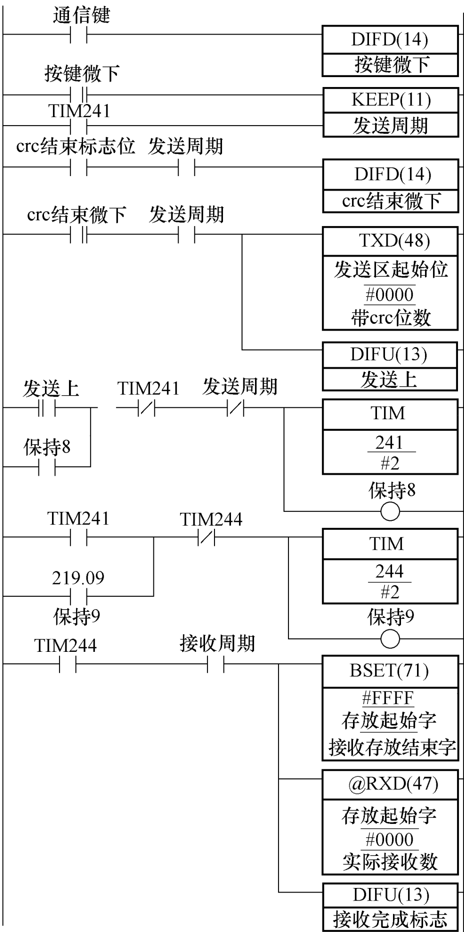


图 7-102　发送、接收程序

据介绍，该程序已在 CPM2A/CPM2AH 与变频器（3G3MV）间进行过多次通信。并通过变频器（3G3MV）的动作及响应验证证明，该程序是可靠的。且得到了熟悉变频器（3G3MV）的技术人员检查、确认。此程序的更详细代码，可从 OMRON 技术支持网站上下载。

7.6.2　用协议宏通信

协议宏通信是指，先在计算机上用集成在 CX-one 中的 CX-Protocol 软件创建的通信协议序列。然后，计算机与 PLC 联机，把这个序列下载给 PLC。PLC 再执行协议宏指令。这时 PLC 将按设计好的这个序列与通信对象通信。

实现协议宏通信要有一定的硬件条件。除了要使用能支持该功能设备，如使用 C200HE 的 COM04～COM06，还必须作好相应的设定。如在 C200HE 上，使用 COM06，要将其 DIP 开关 SW1 调整到“2”侧，终端电阻开关 SW2 设定为 ON。另外，C200HE 的数据区 DM6555、DM6556 要设置成支持协议宏。并且通信连接的格式、使用的协议及通信参数，如波特率等，要与通信对方的设定一致。

CX-Protocol 软件的功能很强，有以下特点：软件支持对话式菜单，使通信序列易于设计；每个协议最多允许定义 1000 个通信序列，即 000～999，每个序列最多允许定义 16 步；每个协议可定义监视时间，响应的应答方式及链接通道；每一步可设定重复次数、发送/接收的数据的信息、下一步处理及出错处理；对于发送和接收数据的地址可任意指定；在发送

和接收的数据信息中自带了许多种校验方式（LRC、CRC、CRC-16、SUM）可由用户选定(用户不必编程)，并在发送时自动添加；在软件内部自带了多种用于与 OMRON 外围设备通信的协议，可按需调用；通过软件还可监视串口交换的数据。

通信序列每一步的结构及各参数的含义如表 7-21 所示。

表 7-21 序列参数的含义

参数	功　能	参 数 设 置
Repeat	设置重复步的次数	常数,IR/SR,LR,HR,AR,DM 和 EM 区域
Command	设置通信命令	发送,接收或发送与接收
Retry	设置在执行发送和接收命令发生错误时,重新执行次数	0 ~9
Send Wait	设置在发送期间等候发送数据的时间	单位 0. 01s,0. 1s,1s 和 1min
Send Message	设置用于接收命令或发送和接收命令的发送数据	识别码,地址,长度,数据,错误检查码和终止符
Receive Message	设置用于接收命令或接收和发送命令的期望接受数据	识别码,地址,长度,数据,错误检查码和终止符
Array	设置用于接收命令或接收和发送命令的期望接受数据(最多 15 种类型),并按数据类型调整处理	识别码,地址,长度,数据,错误检查码,终止符和下一步处理
Response	设置是否写接收数据	是/否
Next	设置当前步顺利结束时转往的下一步	END,GOTO,NEXT,ABORT
Error	设置当前步出现错误时转往的下一步	END,GOTO,NEXT,ABORT

在序列各项参数中，Send Message 和 Receive Message 是最重要的。其内容及格式应与通信对象的通信协议相协调。否则无法通信。

Send Message 和 Receive Message 结构如下

Message Name Header（h）、Terminator（t）、Check Code（c）、Length（l）、Address（a）及 Data（d），其中（h）、（t）、（c）表示信息开始位、结束位、校验位。它们是由通信协议所决定的。当设置（t）时，（l）自动附加，（a）是指信息送往目标的标志符，（d）用于设置信息内容。

以下就某港务局与浙江大学合作，对其下属的煤运码头门机电气传动部分的改造为例，对 Address（a）和 Data（d）作进一步解释。

原有的门机传动分为 3 块，即门机抓斗的 3 个自由度：起升、变幅和旋转，均采用交流绕线式电动机进串电阻调速。其缺点是：能耗大，运行时机械、电气冲击大，故障频繁，维修任务繁重等。故决定将系统改造为 PLC 控制的交流变频调速系统。

该系统组成是：OMRON-C200HE 型 PLC，英国 CT 公司的通用变频器及 OMRON-NT620S 型触摸屏。遵循生产操作习惯，采用手柄操作，触摸屏仅用来显示相关信息，为系统维修提供支持。

为了实现通信，C200HE 配备具有 RS485 口的通信卡。并也给每台 Unidrive 变频器配备了 CT 公司的 UD-71 通信模块（提供有 RS232 和 RS485 光隔通信接口）。本系统全部使用 RS485 接口，组成 RS485 网络进行通信。整个系统配置如图 7-103 所示。

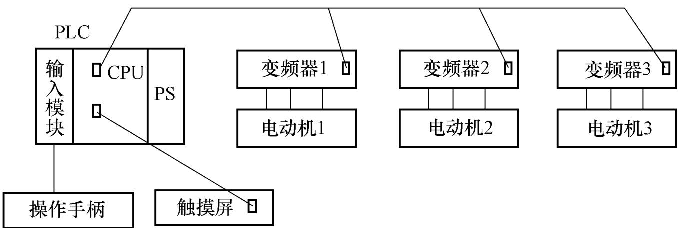


图 7-103　系统配置

从图知，PLC 是该系统控制核心。它根据操作要求和内部逻辑关系，向变频器发送宏通信命令，再从变频器读回各种信息，并将这些信息送给触摸屏显示等。

为了用宏读写 3 台变频器，必须先将地址和参数发给相应的变频器。实现此过程分两步：

第 1 步：编写通信协议。关键的问题在于 Send Message 和 Receive Message 中 Address 和 Data 的编写。

首先需设置它们的属性，有读（R）或写（W）。显然，Send Message 中发送的地址号和参数号是从 PLC 的 DM 区中读出后发送的，所以其属性设为 R；而 Receive Message 中接收到的信息是要写入 DM 区的，所以其属性设为 W。

第 2 步：从指定字中读写地址或数据。有几种方法可以指定该字，一种通用的方法是用包括变量 N 的一阶方程用于地址或数据的引入，每当在通信序列的步中，指定的重复计数器重复一步时，变量 N 加 1，使用带 N 变量的算式计算地址或数据，即可实现地址和数据的动态传输。

根据 CT 变频器通信协议，读变频器参数信息时，每次先发送长度为 8 个字节的地址和参数，返回的信息长度不定。这里设定，每 16 个字节存放 1 条信息，数据长度由（t）确定，后自动附加。Data 中以通配符 * 表示。

由此编制 Send Message 和 Receive Message 如表 7-22 所示。

表 7-22　编制 Send Message 和 Receive Message

* Message Name	Header	Terminator	CheckCode Length	Address	Data
Send1	EOT	ENQ	略	(R(8N),4)	(h)+(a)+(R(8N+4),4)+(t)
Recv1	STX	EXT	略	(W(16N),4)	(h)+(a)+(W(16N),*)+(t)

用上面的通信协议（序列号设为 1），假设发送信息存在 DM300 开始的数据区，接收信息存入 DM800 开始的数据区，则用图 7-104 所示的程序就可连续读取 3 台变频器的指定参数。

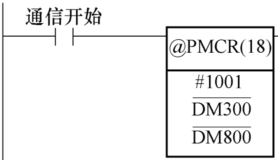


图 7-104　连续读取数据

像这样用一条通信协议宏指令，实现遵循同一个协议的多条信息传送，将简化 PLC 的编程。

值得注意的是，当要执行多个 PMCR 指令时，可让第一个 PMCR 指令执行一段时间后，停止执行。再让第二个 PMCR 指令执行。依此类推。为此，尽量使@ PMCR 微分执行，还要通过定时器，控制各宏指令的依次执行。定时时间可根据客户要求定义。另外，还要利用好通信标志位，如当 28908 通信忙位 ON 时，经一定延时，可用 28911 通信复位位 ON 对其复位，并用它的执行一次协议宏指令。

Unidrive 变频器与上位机之间的通信命令细节属于专门问题，这里就不具体介绍了。

提示：CJ、CP 型 PLC 协议宏指令有 4 个操作数，除了控制字、数据源及数据目标地址，要指定通信端口。

7.6.3 用从站地址通信

当智能装置与 PLC 联网，成为 PLC 的一个从站点时，将具有具体站点地址。PLC 可用对这个地址。用从站地址通信，就是利用这个地址的读写，实现通信。条件是 PLC 必须具有主站功能。而智能装置也必须有联网条件。

如 OMRON 3G3EV 系列的变频器与 PLC 进行通信，如 PLC 具有 SYS BUS 主站功能，就可以上远程 I/O（即 SYS BUS）总线。用读写从站的方式，与变频器通信。

再如 3G3FV 系列，则可选用通信卡单元，上 CompoBus/D 网络。此时可通过 Remote I/O 和 Message Service 两种通信方式监控到变频器参数。

结语

PLC 通信编程是 PLC 编程的重要方面。因为为了增强 PLC 的控制能力，扩大控制规模，实现控制的远程化、信息化，PLC 联网已越来越普遍了。

PLC 联网通信程序类型很多，不同类型的程序应按各自的特点编写。

通信程序具有相关性、交互性、从属性、时延性、可靠性及安全性的特点。所以编写此类程序，常要考虑到网络的实际，要遵守网络协议。同时，注意交互作用，通信双方都要考虑到。此外，还要考虑与整体程序的配合。

编写通信程序涉及的面积较广。除了应具备 PLC 的编程知识，还要熟悉计算机编程软件。

请想想

1. PLC 联网、通信的目的？
2. PLC 联网、通信的类型？
3. PLC 与 PLC 通信可行方案？
4. PLC 与计算机通信可行方案？
5. PLC 与智能装置通信可行方案？
6. 计算机通信编程要点？
7. 通信协议含义？
8. 自由协议通信要点？

请试试

1. 编写链接通信程序
2. 编写指令通信程序
3. 编写协议通信程序
4. 其它实际控制编程

第 8 章　PLC 控制可靠性程序设计

以上各章已讨论了 PLC 的控制、数据采集及通信的程序编写，本章将讨论 PLC 控制可靠性有关问题。

PLC 控制系统可靠性是一个综合的问题，既牵涉到 PLC 自身，也涉及到被控制对象，既牵涉到硬件，也涉及到软件，比较复杂。

本章讨论只集中在编程上，讨论程序执行中的异常处理及编写一些附加程序，以提高系统的可靠性。

8.1　概述

关键词：可靠性、PLC 可靠性、PLC 控制系统、平均故障间隔时间、平均故障修复时间

8.1.1　PLC 控制可靠性概念

1. PLC 可靠性

可靠性是指元件、器件、设备或系统，在规定的条件下与规定的时间内，所能完成的规定功能的能力。而规定的条件、规定的时间及规定功能与具体的对象有关。随之也有不同的量化指标。

PLC 是可修复的产品，常用平均故障间隔时间（Mean Time Between Failure，MTBF）及平均故障修复时间（Mean Time To Repair，MTTR）评价它的可靠性的指标。

因为 PLC 使用的有效度（Availability，A）与两个时间是密切相关的，见下式：

$$A = MTBF/(MTBF + MTTR)$$

式中　A——有效度；

MTBF——平均故障间隔时间；

MTTR——平均故障修复时间。

显然，A 值越大越好，它可使 PLC 系统得到充分的利用。而从上式可知，MTBF 越大，MTTR 越小，则 A 越大。所以，PLC 的可靠措施都是围绕提高 MTBF 及降低 MTTR 值进行的。

增加平均故障间隔时间，主要是厂家的工作。厂家不断地改进 PLC 的硬件及操作系统，取得了很大进步。像 CPU 这样复杂的模块，现在，这个时间已从 5 万小时提高到近 70 万小时。内存模块甚至可达 120 多万小时。这是很可观的。

增加平均故障间隔时间，对用户而言，主要是严格按规定的条件，正确使用 PLC。确保达到厂家规定使用时间。

为减少平均修复时间，PLC 厂家也作了很多努力，已为用户创造了很多条件。用户则应充分利用好这些条件，减少平均修复时间。

例如，PLC 有很多指示灯及存于内部的出错标志。如出现故障，可利用它们判断故障原

因，以至于故障定位。

再如，当出现故障时，有的 PLC 有故障纪录，可用其了解故障出现的时间、条件，进而判断故障原因，以至于故障定位。

一旦故障原因清楚，故障就能定位到模块，所以维修是很简单的。只要更换有问题的模块就可以了。

所以，PLC 自身的可靠性是有保证的。对用户讲，正确使用 PLC，充分利用厂商提供的可靠性的保证条件也就够了。

2. PLC 控制可靠性

PLC 控制可靠性是指 PLC 与其被控对象结合，组成系统的可靠性。系统的目的也是在规定的条件下与规定的时间内，所能完成的规定功能。

PLC 可靠不等于 PLC 控制系统可靠。PLC 控制系统比单独的 PLC 复杂。系统复杂，出现故障的可能性就大。可靠性将降低。

PLC 控制系统不是 PLC 及其被控对象简单地相加，而是两者有机的结合。结合得好，才可靠。

PLC 控制系统也不等于 PLC 及其被控对象全面结合，只是与实施控制有关的才需要结合。主要是系统的传感器，操纵器。前者与 PLC 的输入有关，后者与 PLC 的输出有关。

结合好的关键一是系统要配置好，要提高构成系统的各元器件本身及协调工作的可靠性，另一是程序要设计好，充分利用好硬件条件，确保系统可靠。

系统配置主要是硬件问题。程序设计则是这里要介绍的内容。设计程序，以确保系统可靠，是我们追求的目标。

为了系统可靠，PLC 控制必须可靠，必须做到 3 点：

（1）尽量避免出故障：用可靠的硬件配置及程序设计，避免系统出故障，以增加系统的平均无故障间隔时间。

（2）避免灾难性故障：即使系统出故障，应通过应急程序控制这个故障。力求避免故障扩大，以避免出灾难性故障。如故障能得以控制，将减少排故时间。

（3）便于进行故障诊断：出了故障要有故障纪录，类似飞机上的“黑匣”。以便于人工进行，或能自动进行故障诊断。确保故障准确定位，便于故障修复，减少排除故障时间。

提示：PLC 故障，也称错误，或出错。以下讨论，这几个词基本上是对等的。

8.1.2 PLC 控制可靠性类型

PLC 控制可靠性类型主要指，在编程上应从哪几个方面入手提高 PLC 控制系统的可靠性。

1. PLC 自身工作可靠性

PLC 自身工作可靠是 PLC 控制可靠的关键。没有它自身工作可靠，系统可靠是不可想象的。

PLC 自身工作的编程工作量不大。主要是按要求作好必要的设定，能用工具软件查看有关错误记录，能了解与使用有关错误的代码等等。

再就是利用好有关错误处理指令、错误的代码，进行错误诊断。

2. 输入程序可靠性

无数的实践证明，控制系统的错误多数是由输入错误引起的。传感器错误，输入断线，输入信号受干扰，等等，都会引起输入错误。

输入程序可靠性设计就是，用它的程序去应对可能出现的输入异常、错误，以确保系统安全。

3. 输出程序可靠性程序设计

输出错误也PLC控制系统常见的一种错误。输出程序可靠性程序设计就是用相应程序，去避免这些错误，或出现错误时，还能确保系统安全。

4. 通信程序可靠性程序设计

如果系统是联网的，网络是否正常，通信是否无误，这些都是系统可靠的重要方面。为此，就要设计可靠性很高的通信程序来确保通信数据无误。

同时，还需要通信错误的应对程序，以便出现错误时，不致系统瘫痪，保证系统工作可靠。

再有，当网络开放时，通信的权限也必须管理，不能什么人都可读写数据。

5. 主体程序可靠性程序设计

除了输入、输出及通信程序可靠，主体程序也要可靠。而且，还要尽量在错误预测、错误避免、错误控制、错误纪录、错误报警、错误诊断、容错操作、掉电保户及应急处理方面，也有相应的处理程序。

以上几方面编程将在以下各节分别讨论。

8.1.3　PLC控制可靠性意义

计算机软件的可靠性最初仅仅被认为是软件的准确性。如果软件能够准确无误地完成所要求的功能，人们就认为软件是可靠的。然而，这最起码的要求也常常不能得到满足。有人做过统计，对于初次编出的软件（指计算机软件），平均每100～4000条指令就会出现一个错误。这些错误需要在调试、联调、试运行，甚至到运行时才能陆续被发现和改正。

近年来，人们对软件可靠性赋以更广泛的含义，对计算机软件而言，即便于使用和便于扩展。如果一个软件不便于使用、不便于扩展，就认为这个软件存在着缺陷。

软件的质量主要由以下六方面的因素决定：

（1）时间因素　与硬件一样，软件也有MTBF、MTTR等指标。除此之外，还有以下时间指标：系统平均不工作间隔时间（Mean Time Between System Downs，MTBD）、平均停机时间（Mean Down Time，MDT）。

（2）缺陷频数　包括软件缺陷数、文件缺陷数、用户提出的补充要求数。

（3）与软件可靠性有关的百分率　除了与硬件相似的可靠性、可用性、可维修性、错误率等百分率之外，还有以下几种百分率：

1）不合格率：不能算错误，但应进行改进的事件叫不合格事件，它的出现率叫不合格率。

2）延迟率：一项要求在规定时间 T 内完成的任务，由于软件不可靠，实际完成时间为 T1，则定义 D = T1 − T 为延迟时间，D/T 叫延迟率。

3）误操作率：这与操作者的操作水平有关，但在一定程度上反映了软件说明书是否清楚，以及软件是否适用于操作。

4）原因不明率：出现了软件错误但查不出原因，从而无法纠正的错误率叫原因不明率，它反映了软件的可维修性。

5）同错误事件率：第一次出现的错误在采取措施纠正后，仍重复出现的再现率叫同错误事件率。它反映了纠正措施不彻底。

6）可靠性经费率：为软件可靠性及维修所付出的费用与总软件费用之比叫可靠性经费率。

（4）软件投入　包括开发软件所消耗的工日数或工时数，对软件检查的项目数，对用户提出的要求采取对策的费用等。

（5）软件特性　包括软件是首次开发的还是基本套用的、复杂程度、标准化程度、结构及规模大小、软件的寿命周期等。

（6）使用方面特性　软件使用方面的特性。

以上讲的是计算机软件。对 PLC 而言，可靠性则更不仅仅是“准确性”了，也不仅是“便于使用”、“便于扩展”，这些仅仅是自身的可靠性问题，还应有更高的要求。尽管 PLC 控制可靠性的关键是硬件，但软件的作用也不可忽视。PLC 软件（程序）还应在硬件出现故障时，能检测到故障，能报警故障，能避免故障造成的不良后果，以至于有故障还能“带病”正确工作。即 PLC 软件的可靠性还承担着协助提高硬件可靠性的任务。

事实上，完全可用 PLC 程序实现：检测、报警与避免输入、输出及通信故障及相应后果；一旦出现故障，能及时、准确地记录故障类型、出现时间及部位，为错误诊断提供支持；一旦出现故障，能及时采用应急措施，控制故障扩大，确保人身及设备安全；在未出现故障前，能依据硬件工作状态，预测可能出现的故障；等等。

显然，如果能实现以上目标，对提高 PLC 控制的可靠性将具有重大与积极意义。

8.2　PLC 自身工作可靠性

关键词：致命错误、非致命错误、错误代码、错误信息、FAL 指令、FALS 指令、自定〖错误、错误信息清除

8.2.1　PLC 错误类型

PLC 自身工作是非常可靠的，但难免还是要出现错误。从错误的项目看，以 CJ 系列 PLC 为例有：CPU 错误、内存错误、I/O 错误、程序出错、扫描时间超出错误等等。表 8-1 所示为 CJ 系列 PLC 一些主要错误项目及对应的错误代码。PLC 发生错误，此代码将存于辅助区域 A400（4 位数）中。而 C 系列 PLC 存于在 SR253 的低字节（2 位数）中。但当两个或多个错误同时存在时，只是最高（最严重）错误的代码被保存。

表 8-1　CJ 系列 PLC 错误项目

分　类	错误代码	错误名称
致命的系统错误	80F1	存储器错误
	80C0～80C7,80CE,80CF	I/O 总线错误
	80E9	重复编号错误
	80E1	I/O 点数过多
	80E0	I/O 设置错误
	80F0	编程错误
	809F	循环时间太长
	80EA	扩展机架编号重复
非致命的系统错误	008B	中断任务出错
	009A	基本 I/O 错误
	009B	PC 设置设定错误
	00E7	I/O 检验错误
	0200～020F	CJ 系列 CPU 总线单元错误
	0300～035F,03FF	特殊 I/O 单元错误
	00F7	电池错误
	0400～040F	CJ 系列 CPU 单元设置错误
	0500～055F	特殊 I/O 单元设置错误

从表 8-1 知，从错误的分类看有非致命错误及致命错误。出现非致命错误，系统的报警灯闪烁，但 PLC 仍继续运行、工作。出现致命错误，系统的红色报警灯常亮，PLC 停止运行，不工作。CJ 系列 PLC 还有待机错误，即 PLC 只能在编程状态下待机，不能进入运行或监控状态。

表 8-2 所示为 CJ 系列 PLC 出错时 CPU 单元指示灯显示情况。如果手持式编程器或运行 CXP 软件的计算机与 PLC 相连，如 PLC 出错，也将有信息显示。

表 8-2　CJ 系列 PLC 错误时 CPU 单元指示灯显示

指示灯	CPU 出错	CPU 待机	致命错误	非致命错误	通信出错		输出 OFF 位为 ON
					外设	RS-232C	
RUN	OFF	OFF	OFF	ON	ON	ON	ON
ERR/ALM	ON	OFF	ON	闪烁	—	—	—
INH	OFF	—	—	—	—	—	ON
PRPHL	—	—	—	—	OFF	—	—
COMM	—	—	—	—	—	OFF	—

此外，在有关辅助（AR）区中，还存有分类更详细的错误信息。表 8-3 所示为 CJ 系列 PLC 有关编程出错（程序错误）的相应信息（其它项目可参阅有关说明书）。

表 8-3　CJ 系列 PLC 编程出错信息

错误	错误代码（在 A400）	标志和字数据	可 能 原 因	可能的纠正措施
程序错误	80F0	A40109：程序出错标志 A294～A299：程序错误信息	A29513：微分溢出错误在在线编辑中已经插入或删除太多微分指令	在写入程序修改后，切换到 PROGRAM 模式然后再返回到 MONITOR 模式继续编辑程序
			A29512：任务出错出现一个任务错误，下列条件将产生一个任务错误： 1）没有可以执行的循环任务 2）没有将一个程序分配给任务，检查 A294 可以得到丢失程序的任务编码 3）在 TKON（820），TKOF（821）或 MSKS（690）指令中指定的任务不存在	检查起动循环任务属性 检查由 TKON（820），TKOF（821）控制的每一个任务的执行状态 确认在 TKON（820），TKOF（821）和 MSKS（690）指令中指定的任务号码都有相应任务 使用 MSKS（690）掩盖没有被使用和没有设置编程的任何 I/O 口或计划中断任务
			A29510：非法访问错误。出现非法访问错误并且 PC 设置已经设置为出现一个指令错误时停止操作。下列是非法访问错误： 1. 读/写一个参数区域 2. 写没有安装的存储器 3. 写入作为 EM 文件存储器的一个 EM bank 4. 写入一个只读区域 5. 当指定 BCD 模式时，间接 DM/EM 地址不是 BCD 码	找出出现错误（A298/A299）的程序地址，改正指令
		A40109：程序错误标志 A294～A299：程序错误信息	程序错误，详见本表的下面列 错误停止的地址将输出到 A298 和 A299	检查 A295 确定出现的错误类型并检查 A298/A299 找到出现错误的程序地址
			A29511：无 END 错误	确认在 A294 指示的任务中（程序停止的任务编号）有 END（001）指令
			A29515：UM 溢出错误，在 UM（用户程序存储器）溢出	使用编程设备重新传送程序
			A29509：间接 DM/EM BCD 错误 出现一个间接 DM/EM BCD 错误并且 PC 设置已经设置为出现一个指令错误时停止操作	找出出现错误（A298/A299）的程序地址，改正间接地址或改为二进制模式
			A29508：指令错误 出现一个指令处理错误并且 PC 设置已经设置为出现一个指令错误时停止操作	找出出现错误（A298/A299）的程序地址，改正间接地址或改为二进制模式
			A29514：非法指令错误 程序包括一个不能执行的指令	将程序重新传输到 CPU 单元

CP1H 型 PLC 出错也称异常。其 CPU 异常有 4 种，如表 8-4 所示。

表 8-4　CP1H CPU 异常状态

异常状态	内　　容
CPU 异常	CPU 单元发生 WDT(监视定时器)异常,作为 CPU 单元不能动作、停止运行
CPU 待机中	不具备开始运行的条件,故待机
致命错误	由于发生重大的问题,故不能继续运行,停止
非致命错误	发生轻微的问题。继续运行

到底哪种异常可通过工作指示 LED、7 段 LED 及特殊辅助继电器确认。图 8-1 所示为 CP1H 型 PLC 面板。

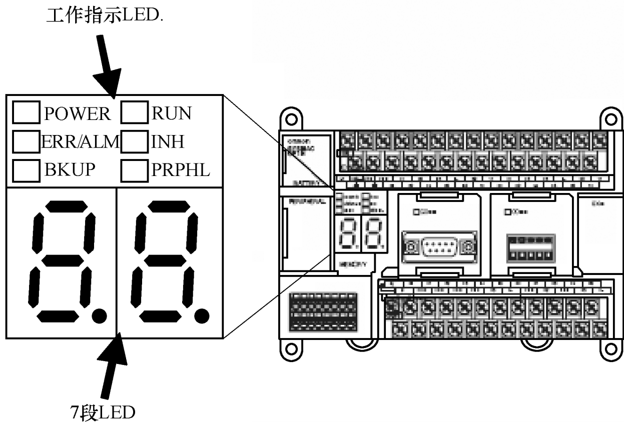


图 8-1　CP1H 型 PLC 面板

从图知，它有 6 个指示灯，可用以监视 CPU 工作。表 8-5 所示的为指示灯与 CPU 状态的对应关系。如果出现异常，从中可初步确认。

表 8-5　指示灯与 CPU 状态对应关系

POWER(绿)RUN(绿)	灯亮	通电时
	灯灭	不通电时
	灯亮	运行或者监视模式下程序执行中
	灯灭	程序模式停止中或者因运行停止异常停止中
ERR/ALM(红)	灯亮	发生运行停止异常,或者发生 CPU 异常(WDT 异常)时停止运行,所有输出切断
	闪烁	发生运行继续异常,此时运行继续
	灯灭	正常时
INH(黄)	灯亮	输出禁止标志(A500.15)ON 时灯亮 所有输出切断
	灯灭	正常时
BKUP *(黄)	灯亮	写入到内部内存中或者访问存储器 另外,PLC 本体的电源再接通时,用户程序复原期间灯也亮
	灯灭	除上述以外
PRPHL(黄)	闪烁	外围设备 USB 接口通信中(收发信执行中)时,灯闪烁
	灯灭	除上述以外

从图知，它还有 2 个 7 段 LED。可显示为 2 位发生异常时故障代码。但故障代码为 4 位，故每次 2 位分 2 次显示。另外异常详细信息也有连续 4 位的情况，则按每次 2 位依次显示。图 8-2 所示为一个异常显示实例。它显示故障代码：80F1（存储器异常）及异常详细信息：0001（用户程序）。

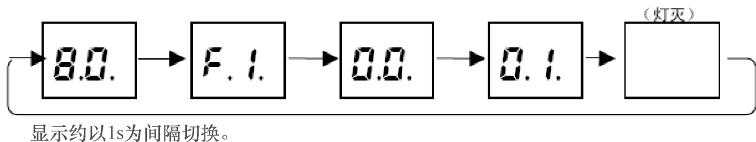


图 8-2 异常显示实例

如果有多个异常同时发生时，重要度高的优先显示。异常解除时重要度次高的再显示。7 段 LED 虽也可根据指令执行显示或显示模拟电位器操作，但发生异常时显示故障代码优先。故障代码的细节请参阅有关说明书。

此外，还可查阅 AR 区确认故障。表 8-6、表 8-7 所示的分别为 CP1H 型 PLC 致命与非致命错误时的有关信息。

表 8-6 CP1H 型 PLC 致命错误信息

状态	故障代码（A400 CH）	异常标志	异常内容继电器	
			异常内容	地址
存储器异常	80F1	A401.15	存储器异常发生地点	A403 CH
I/O 总线异常	80C0～80C7，80CA，80CE，80CF	A401.14	I/O 总线异常详细信息	A404 CH
单元号重复使用错误	80E9	A401.13	重复 CPU 总线单元编号	A410CH
			重复 No.	A411～A416 CH
I/O 点数过多	80E1	A401.11	I/O 的特殊 I/O 单元点数过多详细信息	A407 CH
I/O 设定异常	80E0	A401.10	—	—
程序错误	80F0	A401.09	程序错误详细信息	A294～A299 CH
周期时间过长	809F	A401.08	—	—
FALS 指令异常	C101～C2FF	A401.06	—	—

表 8-7 CP1H 型 PLC 非致命错误信息

状态	故障代码（A400 CH）	异常标志	异常内容继电器	
			异常内容	地址
FAL 指令异常	4101～42FF	A402.15	执行 FAL 编号	A360～A391 CH
Flash memory 异常	00F1	A315.15	—	—
中断任务异常	008B	A402.13	中断任务异常发生机号单元号	A426 CH

（续）

状　　态	故障代码(A400 CH)	异常标志	异常内容继电器	
			异常内容	地址
PLC 系统设定异常	009B	A402.10	PLC 系统设定异常位置	A406 CH
内置模拟量异常	008A	A315.14	内部模拟量异常信息	A434 CH
CPU 总线单元异常	0200～020F	A402.07	异常单元编号	A417 CH
特殊 I/O 单元异常	0300～035F,00FF	A402.06	无显示	A418～A423 CH
选项板异常	00D1,00D2	A315.13	异常选项槽编号	A424 CH
电池异常	00F7	A402.04	—	—

8.2.2　PLC 系统错误记录

PLC 多有错误记录区。如 CJ 系列 PLC，每出现一个错误，操作系统将在错误记录区中存储一个错误记录。出错记录包括错误代码（存入 A400），出错内容（出错信息，也用代码表示）和出错时间。最大可以存储 20 个记录。

当记录数超过 20 时，最旧的记录（A195～A199）被删除。最新的记录存在 A100～A104 中。图 8-3 所示为 CJ1 型 PLC 错误记录区存储错误信息的情况。

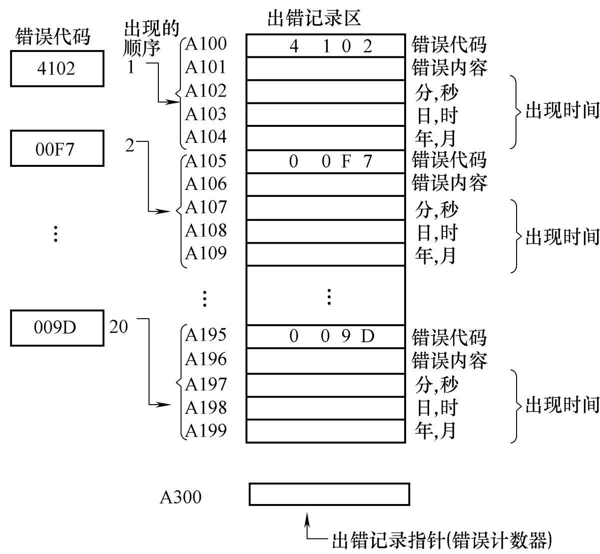


图 8-3　错误记录区存储的错误信息

通过接通出错记录指针复位位（A50014）可以复位出错记录指针，从手持编程器或 CXP 编程软件都可有效的消除出错记录。复位指针不能消除出错记录区的内容。

CP1H 型 PLC 每次发生异常保存在特殊辅助继电器的异常记录存储区（A100～A199 CH）中。另外，除 PLC 的系统异常，用户定义的异常（故障）也可保存。异常记录保存最多也是 20 条。

错误记录可用编程器阅读，也可在 CXP 软件 PLC 错误窗口上查看。图 8-4 所示为在编



图 8-4 在编程软件上看到的出错信息

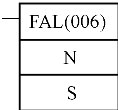
程软件上看到的某出错信息。

8.2.3 PLC 监控指令及其应用

前以介绍，PLC 自身产生错误，PLC 的操作系统会进行错误记录及信息存储。PLC 控制系统的错误也能这么做吗？答案是肯定的。关键是要用 PLC 故障处理指令 FAL（006）或 FALS（007），先定义控制错误。定义后一旦出现这个控制错误，PLC 操作系统也将与处理自身错误一样处理所定义的错误。

1. FAL（006）指令及其应用

用于产生或清除用户定义的错误操作数功能，其格式为



这里的 N、S 的含义及其取值见表 8-8。但 CS1-H、CJ1-H、CJ1M 型 PLC 与此有稍许不同，请参看有关说明。

表 8-8 N、S 含义及其取值

N	S	功　能
0	#0001 ~ #01FF	清除带对应 FAL 号的非致命错误
	#FFFF	清除所有非致命错误
	其它*	清除最严重非致命错误
1 ~ 511（这些 FAL 号与 FALS 共享）	#0000 ~ #FFFF	产生一个带对应 FAL 号的非致命错误（无信息）
	字地址	产生一个带对应 FAL 号的非致命错误 S ~ S + 7 中包含的 16 个字符的 ASCII 信息将显示在编程设备上

注：其它设置是常数#0200 ~ #FFFE 或一个字地址。

从表可知，FAL（006）的操作与 N 的值有关。将 N 设置为 0000 是清除错误，N 为 0001~01FF（即 0 ~ 511）是定义一个错误。

如果 S 是一个字地址，并且在 S ~ S + 7 已存储了 ASCII 信息（最多 16 个字符，不然可用空字符，即十六进制 00，代表信息结束），当 FAL（006）执行时，该信息将显示在编程

设备上。每个字中最左边的字节将先显示。如果不需要信息，可在 S 中设置一个常数。

这两个指令都有逻辑条件，并可微分执行。如果逻辑条件也就是定义的“控制错误”出现的条件。

当执行 FAL（006）且 N 设置了一个与 A529（产生 FAL/FALS 系统号）内容不同的 FAL 号（&1 ~ &511），将产生该 FAL 号的非致命错误，并将执行下列过程：

（1）FAL 错误标志（A40215）将变 ON（PLC 将继续运行）。

（2）对应 FAL 号的执行 FAL 号标志将变 ON：标志 A36001 ~ A39115 对应于 FAL 号 0001 ~ 01FF（1 ~ 511）。

（3）错误代码将写入 A400。对应于 FAL 号 0001 ~ 01FF（1 ~ 511）错误代码是 4101 ~ 42FF。但在 FAL（006）指令执行时，同时发生了一个致命错误或一个更严重的非致命错误，最严重的错误代码将写入 A400。

（4）错误发生的时间和错误代码将写入错误记录区（A100 ~ A199）。但对 CS1-H、CJ1-H、CJ1M CPU 单元，如果 PLC 设置以使 FAL（006）产生的错误不被记录，即如果手持编程器地址 129 位 15 设为 1，错误记录不会写入错误记录区。

（5）CPU 单元上的 ERR 指示灯将闪烁。

（6）如果在 S 中指定了一个字地址，以 S 开始的信息将被记录（显示在编程设备上）。

当控制错误排除、定义错误出现的条件不存在时，还必须用 N 为 0，再执行一次 FAL 指令，才能清除这个错误，使系统恢复正常。

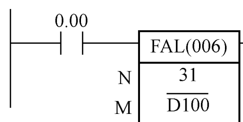


图 8-5　定义非致命错误程序

例：定义一个非致命错误及错误清除。

定义非致命错误程序如图 8-5 所示。这里 000000 ON，即为控制故障出现条件。

从图知，N 为 31，即为错误号。M 设为地址 D00100。从 D00100 ~ D00107 中的 ASCII 码如图 8-6 所示。

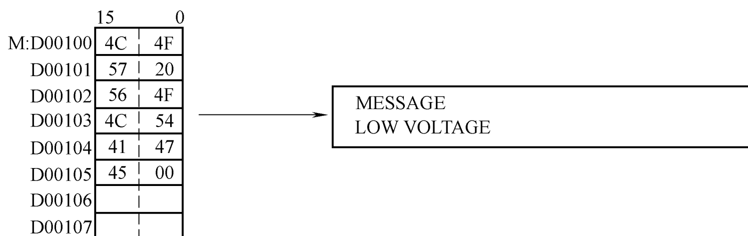


图 8-6　M 中信息内容

可知，这里当 CIO 000000 ON（意味着出错），执行 FAL（006），将产生一个 FAL 号为 31 的定义非致命错误，并将执行下列过程：

（1）FAL 错误标志（A40215）将变 ON。

（2）对应的执行 FAL 号标志（A36114）将变 ON。

（3）错误代码（411F）将写入 A400。

（4）错误发生的时间、代码和错误代码将写入错误记录区（A100 ~ A199）。

（5）CPU 单元上的 ERR 指示灯将闪烁。

（6）D00100～D00107 中的 ASCII 信息（LOWVOLTAGE）将显示在外围设备上。

图 8-7 所示为错误清除程序。

当 000000 OFF，而 000001 ON，执行 FAL（006）将清除一个 FAL 号为 31（即 M 为#001F）的非致命错误，使相应的已执行 FAL 标志（A36114）变 OFF，并使 FAL 错误标志（A40215）变 OFF。

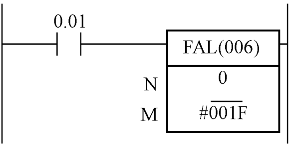


图 8-7　错误清除

对 C 系列 PLC，这个指令只有一个操作数 N（一个字节），用以设定错误代码。代码在 1～99 间任选。一旦执行本指令，这个代码将存入特殊继电器 SR253 通道低字节中。

总之，有了定义错误记录及历史错误记录，将大大方便了 PLC 及其控制系统的错误诊断。

2. FALS（007）**指令**

它与 FAL 指令不同的只是，它定义的是致命错误。一旦执行它，系统将不能工作，停止程序运行。

而清除这个错误，除了错误条件不再存在，系统还要复位，重新起动。故定义致命错误要慎重。实在需要处理控制出现的致命错误，可使用别的办法。

3. WDT（94），**SCAN**（18）**指令**

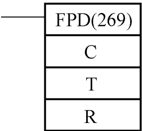
用于控制与监视程序扫描时间。

WDT 用以设定要延长的程序扫描时间。有的最多可延长到 6500ms。一般情况下，PLC 程序扫描时间超 100ms 即报警，超过 130ms PLC 可能要被迫停止工作。但若用户程序很多，必须有长的扫描时，则可用它予以延长。

SCAN 用于设定最小的扫描时间。若 PLC 扫描时间少于它时，将等待。若大于它，则按实际执行。最小设定时间为 0～9999ms，常用于须要定时运行程序的场合。

4. FPD（269）**指令**

监视执行 FPD（269）和执行诊断输出之间来诊断指令块中的错误，并查找哪个输入阻止输出变 ON。指令格式为



这里 C——控制字。必须是 0000～01FF 之间或 8000～81FF 之间的常数。控制字中各位的功能如图 8-8 所示。

T——监视时间。必须在 0000～270F（十进制 0～9999）之间。T 值为 0，禁止时间监视；1～270F 对应监视时间 0.1～999.9s。

R——信息存储区首地址。

FPD（269）用于进行时间监视和逻辑诊断。图 8-9 所示为该指令执行的情况。

如图，当执行条件 A ON，FPD 指令执行，即计时。如果在 T 指定的监视时间内，输出 B 没有变 ON，将产生非致命错误，并使进位标志变 ON。同时，逻辑诊断将指出，在执行条件 C 中哪个输入阻止输出 B 变 ON。这里输出 B 是在逻辑诊断块中，是从 FPD 之后的第一个 LD（不是 LD TR）或 LD NOT 指令开始，在第一个 OUT（非 OUT TR）或其它输出指令

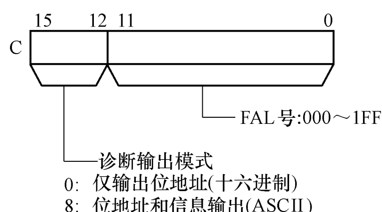


图 8-8　控制字中各位的功能

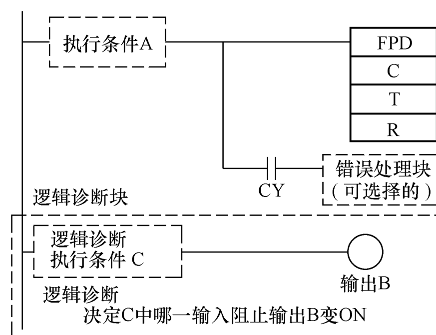


图 8-9　FPD 指令执行情况

结束。

当进位标志变 ON 时，将发生以下过程（如果在 C 中的 FAL 号设置为 000，不执行该过程）：

（1）FAL 错误标志（A40215）将变为 ON（PLC 继续运行）。

（2）指定 FAL 号的执行 FAL 号标志变 ON（标志 A36001 ~ A39115 对应 FAL 号 001 ~ 1FF）。

（3）相应的错误代码将写入 A400。错误代码 4101 ~ 42FF 对应 FAL 号 001~ 1FF。如果同时发生了一个更严重的错误（具有更高的错误代码），更严重错误代码将写入 A400 中。

（4）错误代码和错误发生的时间/日期将写入错误记录区（A100 ~ A199）。

（5）CPU 单元上的 ERR 指示灯将闪烁。

（6）如果输出模式设置成位地址和信息输出（C 中最左位数设置成 8）。R +2 ~ R +9 中存储的 ASCII 信息将作为非致命错误信息显示。

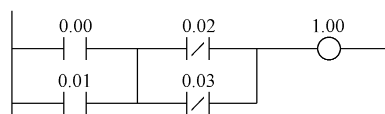


图 8-10　逻辑诊断例

图 8-10 所示的为逻辑诊断的例子。

如图所示，如果输入位 0. 00 ~ 0. 03 全为 ON，FPD（269）将决定常闭条件 0. 02 导致输出 1. 00 保持位 OFF。这样一来，使用位地址找到标志（R 的位 15）变 ON，并将位地址写入信息存储区 R +2 ~ R +4 中。

只要 FPD（269）的执行条件为 ON，逻辑诊断功能每个循环都执行。逻辑诊断功能的运行独立于时间监视功能。

当有两个或更多的输入位阻止诊断输出变 ON 时，执行条件中第一个输入位（最上面的指令行和最接近左边母线）的地址将输出到 R +2 ~ R +4 中。

逻辑诊断功能检查 LD，LD NOT，AND，AND NOT，OR 和 OR NOT 指令的输入位（包括微分和立即刷新变化），而不检查其它指令的输入位和通过索引寄存器间接寻址的操作数。

用 C 的最左位数字可以设置两个诊断输出模式。

（1）位地址输出模式（C 的最左位数字 =0）　当查找到输入位地址后，R 的 15 位（位地址找到标志）变 ON，R 的位 14 指明输入是常开还是常闭。输入位的 8 位十六进制内部 I/O内存地址输出到 R +3 和 R +2 中。

（2）位地址和信息输出模式（C 的最左位数字 =8）当查找到输入位地址后，R 的 15

位（位地址找到标志）变 ON，R 的位 14 指明输入是常开还是常闭。输入位的地址将以 6 个 ASCII 字输出到R +2 ~ R +4 中。

例：使用 FPD 指令及所诊断的输出（200.00）梯形图程序如图 8-11 所示。

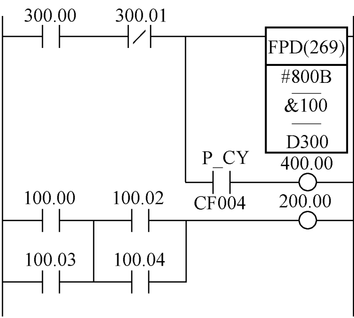


图 8-11　最左位为 0 时信息区内容

从图知，如果 300.00 和 300.01 都变 ON 之后的 10s 内，若 100.00 及 100.03 均 OFF，造成输出（200.00）不变 ON，将产生一个非致命错误，并执行以下过程：

（1）进位标志变 ON。相应将使 400.00 ON。

（2）当 C 的最右 3 个数字指定 FAL 号为十六进制 000B（10），则相应的已执行的 FAL 号标志（A36011）变 ON，相应的错误代码（410B）写入 A400，且 FAL 错误标志（A40215）变 ON。

由于 C 的最高位为 1（位地址和信息输出模式），100.00（因在梯形图中，它处于的位置比 100.03 上，故认定它为首要原因）的地址以 ASCII 形式输出到 D00302 和 D00304 中。具体存储值如图 8-12 所示。

R:D00300	I O		
R+1:D00301	不使用		
R+2:D00302	30	31	包含 ASCII 形式的位地址（010000 转换成 ASCII）
R+3:D00303	30	30	
R+4:D00304	30	30	
R+5:D00305	2D	30	
R+6:D00306	54	25	用户设置的FAL错误信息通过时间功能输出到外围设备,外围设备将显示以下信息:010000-0 ERROR
R+7:D00307	25	F4	
R+8:D00308	25	00	
R+9:D00309	00	00	
R+10:D00310	00	00	

图 8-12　具体存储值

从图知，这里 R 最高位（BIT）为 1。R +2 ~ R +4 为 0100.00 的 ASCII 表示的地址。而 R +6 为，R +6 ~ R +10 位如出错，用户准备加在外设上显示的信息。

C 系列机也有此指令，功能与此相同，但具体的规则、地址稍有差别。另外，这里的时间 T，还可通过示教设定，具体见有关说明书。

8.3　PLC 输入程序可靠性

关键词：输入信号出错、防抖动、数字滤波、非法输入防止及报警、输入冗余及出错报警、输入容错及出错报警

虽然 PLC 有很高的可靠性，但如果输入信号出错，模拟量输入偏差较大，这些都可能使控制出错，造成损失。有人统计，一些自控系统的错误来源几乎 90% 以上就是它。

1. 输入信号出错

输入出错常与输入元器件、接线及信号受干扰有关，如：开关或继电器的机械触点接触不良或抖动；变送器不能正常工作或偏差大；传输信号线短路或断路，现场信号无法传送给 PLC；现场干扰严重，信号失真，等等。

2. 防输入出错处理方法

防止输入出错，有很多硬件的处理办法，如输入受干扰严重，可采取如下措施：

（1）变频器和 PLC 分别接地；

（2）动力线和信号线分开接；

（3）把变频器的载波频率设低些；

（4）在输入点 COM 端，接一个 0.1μF 的电容；

软件方面，有防抖动、数字滤波，非法输入防止、输入冗余及输入容错的功能。

（1）防抖动或输入冲击。一般讲，PLC 的输入信号都接有滤波器，以防接点抖动或冲击，滤波时间常数为 8ms，即只有当输入信号作用 8ms 以上，才算有了输入。这个值一般还可通过对 PLC 设定作改变。

有时，通过设定还满足不了要求，也可用程序进行滤波。办法是用定时器。图 8-13 所示即为一例。

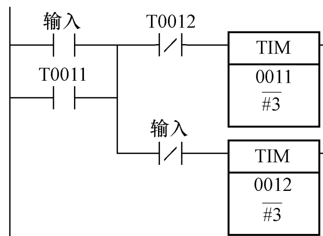


图 8-13　输入信号滤波

从图 8-13 知，当“输入”ON 时，只有超过 300ms，TIM0011 才能 ON，并自保持。接着，当“输入”OFF 时，也只有超过 300ms，TIM0012 才能 ON，进而才能使 TIM0011 OFF。显然，TIM0011 常开接点的通断，就相当滤波后的“输入”的通断。这里，接通进行滤波，断开也进行滤波。

（2）防输入脉冲丢失。采集脉冲量应避免丢失脉冲。其办法有：

1）用高速计数功能采集，只要确保高速计数最高工作频率高于脉冲频率，就不会丢失脉冲；

2）用定时中断及脉冲采集子程序采集，只要确保采集时间间隔小于脉冲频率的倒数，就不会丢失脉冲；

3）用外中断（有中断功能的输入点）及脉冲采集子程序采集，也要确保中断响应速度足够高，才不会丢失脉冲。

如果脉冲频率不高，如每秒在 10 次以下，一般的输入点也可进行采集。

（3）数字滤波。若采集的为模拟量，由于多数 A/D 单元自身有滤波功能，有的还可求平均值，作预处理。故进入 PLC CPU 的已是排除干扰后的信号。

有的 A/D 单元求平均数，如 OMRON 的 CPM1A_MD002，可用以滤波，可防干扰。其效果是较明显的。

若 A/D 单元没有防干扰措施，也可通过程序进行数字滤波，克服干扰。滤波的方法很多，可以求平均值，也可加权平均。此外还有中值法、抑制脉冲算术平均法、一阶惯性滤波法、程序判断滤波法和递推平均滤波法等。

有人总结出经典的数据滤波方法有 10 种：

1）限幅滤波法（又称程序判断滤波法）。根据经验判断，确定两次采样允许的最大偏差值（设为 A）。每次检测到新值时判断：如果本次值与上次值之差小或等于 A，则本次值有效；如果本次值与上次值之差大于 A，则本次值无效，放弃本次值，用上次值代替本次值。

这种滤波的优点是，能有效克服因偶然因素引起的脉冲干扰；缺点是，无法抑制周期性的干扰，平滑度差。

2）中位值滤波法。连续采样 N 次（N 取奇数），把 N 次采样值按大小排列，取中间值为本次有效值。

这种滤波的优点是，能有效克服因偶然因素引起的波动干扰，对温度、液位的变化缓慢的被测参数有良好的滤波效果；缺点是，对流量、速度等快速变化的参数不大好用。

3）算术平均滤波法。连续取 N 个采样值进行算术平均运算。N 值较大时，信号平滑度较高，但灵敏度较低；N 值较小时，信号平滑度较低，但灵敏度较高。N 值的选择取决于采集对象，一般讲，如采集流量，N = 12；采集压力，N = 4。

这种滤波的优点是，它有一个平均值，信号在某一数值范围附近上下波动，适用于对一般具有随机干扰的信号的滤波。其缺点是，对于测量速度较慢或要求数据计算速度较快的实时控制不大适用，所占用的内部器件较多。

4）递推平均滤波法（又称滑动平均滤波法）。把连续取 N 个采样值看成一个队列，队列的长度固定为 N。每次采样到一个新数据放入队尾，并扔掉原来队首的一次数据（先进先出原则）。把队列中的 N 个数据进行算术平均运算，就可获得新的滤波结果。

N 值的选取：采集流量，N = 12；采集压力，N = 4；采集液面，N = 4 ~ 12；采集温度，N = 1 ~ 4。

这种滤波的优点是，对周期性干扰有良好的抑制作用，平滑度高，适用于高频振荡的系统。其缺点是，灵敏度低，对偶然出现的脉冲性干扰的抑制作用较差，不易消除由于脉冲干扰所引起的采样值偏差，不适用于脉冲干扰比较严重的场合，所占用的内部器件较多。

5）中位值平均滤波法（又称防脉冲干扰平均滤波法）。相当于“中位值滤波法”+“算术平均滤波法”。它连续采样 N 个数据，去掉一个最大值和一个最小值，然后计算 N - 2 个数据的算术平均值。N 值的选取：3 ~ 14。

这种滤波融合了以上两种滤波法的优点，对于偶然出现的脉冲性干扰，可消除由于脉冲干扰所引起的采样值偏差。其缺点是，测量速度较慢，和算术平均滤波法一样，所占用的内部器件较多。

6）限幅平均滤波法。相当于“限幅滤波法”+“递推平均滤波法”。它每次采样到的新数据先进行限幅处理，再送入队列进行递推平均滤波处理。

这种滤波的优点是，融合了以上两种滤波法的优点，对于偶然出现的脉冲性干扰，可消除由于脉冲干扰所引起的采样值偏差。其缺点是，所占用的内部器件较多。

7）一阶滞后滤波法。取 a = 0 ~ 1，本次滤波结果 = (1 - a) × 本次采样值 + a × 上次滤波结果。

这种滤波的优点是，对周期性干扰具有良好的抑制作用，适用于波动频率较高的场合。其缺点是，相位滞后，灵敏度低（滞后程度取决于 a 值大小），不能消除滤波频率高于采样频率的 1/2 的干扰信号。

8）加权递推平均滤波法。是对递推平均滤波法的改进，即不同时刻的数据加以不同的“权”，通常是，越接近当前时刻的数据，这个“权”取得越大。给予新采样值的权系数越大，则灵敏度越高，但信号平滑度越低。

这种滤波适用于有较大纯滞后时间常数的对象和采样周期较短的系统。但对于纯滞后时

间常数较小，采样周期较长，变化缓慢的信号，不能迅速反应系统当前所受干扰的严重程度，滤波效果差。

9）消除抖动滤波法。设置一个滤波计数器，将每次采样值与当前有效值比较：如果采样值等于当前有效值，则计数器清零；如果采样值不等于当前有效值，则计数器加1，并判断计数器是否大、等于上限N（溢出）。如果计数器溢出，则将本次值替换当前有效值，并使计数器清零。

这种滤波对于变化缓慢的被测参数有较好的效果，可避免在临界值附近控制器的反复开/关跳动或显示器上数值抖动。但它不宜用于采集快速变化的参数。而且，如果在计数器溢出的那一次采样到的值恰好是干扰值，则会将干扰值当作有效值导入系统。

10）限幅消除抖动滤波法。相当于“限幅滤波法”+“消抖滤波法”。它先限幅，后消除抖动。

这种滤波具有“限幅”和“消除抖动”的优点。改进了“消除抖动滤波法”中的某些缺陷，可避免将干扰值导入系统。但它不宜用于采集快速变化的参数。

提示：还可能设计更多的软件滤波方法，但最有效的滤波还在硬件。在硬件上，采取有效的防干扰措施，才是可靠的滤波。软件滤波只是硬件防干扰措施的补充。

图8-14所示梯形图为连续采集5次数，并剔除其中最高及最低两个数，然后再对其余的3个数作平均，并以其值作为采集数。这5个数通过5个周期（长度可选定，该图程序周期为1s）进行采集。

该图有4个部分：

1）用移位指令，产生自200.00到200.04轮流为ON的控制位。

2）用200.00～200.04把A/D有关通道的内容（图为100通道），分别存入DM100～DM104中。

3）在200.04 ON时，进行一系列比较（图未详细列出具体指令），把DM100～DM104中的最大及最小数剔除，中间数存于DM110～DM112中。

4）对DM110～DM112中的数求和并除以3，其结果存于DM115中。这里的DM115即为滤波后的值。

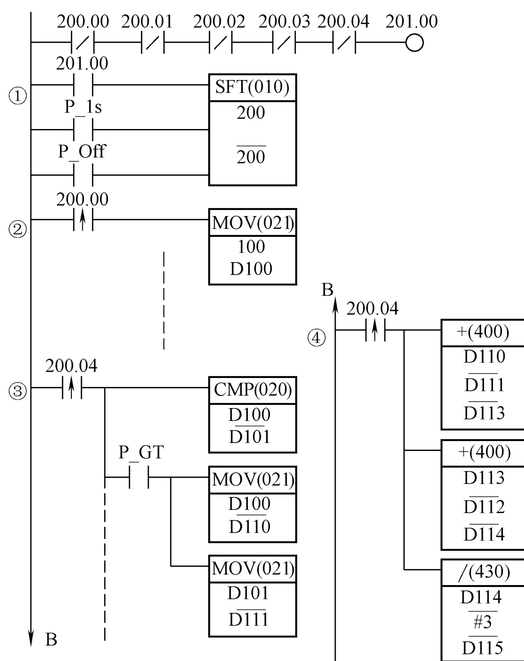


图8-14　滤波程序例

（4）非法输入防止。利用信号之间关系来判断信号是否非法。如左右两个行程开关，绝对不可能两者同时处于ON状态。出现那样状态，即为非法输入，应报警，并禁止其起作用。再如进行液位控制，由于贮罐的尺寸是已知的，进液或出液的阀门开度和压力是已知的，在一定时间里罐内液体变化高度大约在什么范围可是预测的。如果液位计送给PLC的数据和估算液位的高度相差较大，判断可能液位计出错。对这个错误输入应报警，并予以拒绝。

又如各贮罐有上下液位极限开关保护。如开关动作，可先判断这个信号是否正确？可将这信号和该罐液位计信号对比，如果液位计读数也在极限位置，说明该信号是正确的；如果液位计读数不在极限位置，则可能是液位极限开关错误或液位计读数错误。对此，也应报警，并应予拒绝。

再就是误操作避免的设计，如电动机正转、反转两个按钮，如同时按下，就是误操作。这时，应通过输入互锁使 PLC 对此不做任何反应。

也可使用联锁实现误操作避免，如当出现某种情况时，使有的输入被禁止（联锁），以防止非法收入，确保系统安全。

（5）输入冗余与出错检测。传感器有检测逻辑量与模拟量两种。对传感器监控，办法是冗余，用两个传感器。同时从它读信号，然后作比较。看其是否不一致。如不一致超过允许时间，即说明其中之一必有错误。图 8-15 所示为输入冗余及出错报警逻辑。

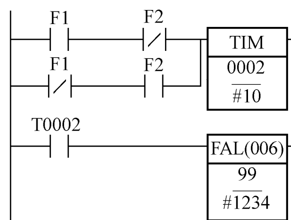


图 8-15　输入冗余及出错报警逻辑

图中 F1、F2 为两个输入开关，用以检测同一对象。正常情况下，其结果应是相同的。所以，T0002 不会工作。若一个出现错误，则 T000 工作，当时间超过设定值（此次定为 1s），则报警，提示出现错误。

模拟量传感器的监控大致也相同。它靠对两个测量值进行比较，视其偏差是否在允许的范围内。若超过允许范围，再看是否处于允许的时间之内。只有超过允许误差范围，而又超过允许的超差时间，才视为出了错误。

（6）输入容错。对重要的输入，必要时也可作 3 选 1 配置，一旦一个出错，一方面报警，另一方面，可按三选二的原则继续工作。当然，这么做的代价是较高的。

8.4　PLC 输出程序可靠性

关键词：输出执行出错、输出监控、误动作避免、掉电保护、应急处理

虽然 PLC 输入信号没有错，模拟量输入偏差也不大，PLC 处理后，得出控制输出也正确，但如果 PLC 输出控制的执行机构没有按要求动作，这些也会使系统出现错误。

为此，在提高输入可靠性的同时，也要提高输出执行动作的可靠性。一旦出现错误，PLC 应及时发现，及时报警。

1. 输出执行错误

输出执行错误与系统的执行机构有关。如：

控制负载的接触器不能可靠动作，虽然 PLC 发出了动作指令，但执行机构并没按要求动作；

控制变频器起动，由于变频器自身错误，变频器所带电动机并没按要求工作；

各种电动阀、电磁阀该开的没能打开，该关的没能关到位。等等。

2. 处理输出执行错误方法

处理输出执行错误的硬件办法很多，在软件上有输出监控及错误输出避免。

（1）输出监控。对执行元件监控有两种方法：一是用“看门狗”监控程序；另一是用

动作反应检测程序。这两个方法本质上是相同的，只是，一个看在给定的时间内动作完成了没有；一个检查两者动作是否一致。

图 8-16 所示为“看门狗”监控的梯形图程序。

从图知，Q 起动动作 5.00 的同时，把 TIM 0000 也同时起动。这里的定时值设为 10s。若 10s 内反馈信号 D 到来，则不执行 FAL 01 指令。不然，将执行 FAL 指令。PLC 的 ALARM（报警）灯闪烁，并于 253（对 C 系列 PLC 而言）通道的低字节记 01（错误号）。如果要把错误的时间、错误号记入数据区，可用数据采集的方法进行。

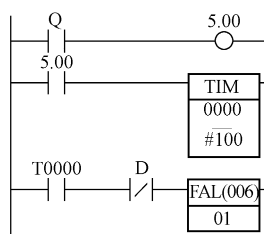


图 8-16　输出监控 1

图 8-17 所示为执行器动作反应检测程序。这里假设用 PLC 输出点 5.00 控制接触器。F 是接触器常开辅助触点连接的 PLC 输入点，用以反映接触器状态。

如图，这里梯形图第 2 梯级，输入条件为 5.00 与 F 的“异或”。只要两者状态一致 TIM 0001 就不会工作；不一致，且时间超过定时设定值则工作，报警。

图 8-16 与图 8-17 的不同是，前者只监视从 OFF 到 ON 时是否及时执行，后者从 OFF 到 ON 及从 ON 到 OFF 都监视。有了这个监视，一旦出了问题，就可知道问题出在哪儿。

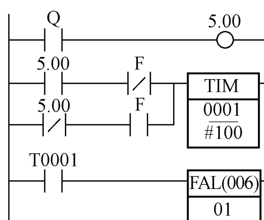


图 8-17　输出监控 2

（2）误动作避免。有时，出现某种输出是不允许的。这时，可把这种输出视为误输出，在逻辑上予以禁止。

8.5　PLC 通信程序可靠性

关键词：通信可靠、数据校验、重复通信、FCS、通信安全

随着通信技术的进步，PLC 与 PLC，PLC 与计算机，PLC 与可编程终端及 PLC 与智能装置可靠通信是有硬件保证的。但由于通信分布广，易受到种种干扰。通信错误或出错也是难免的。故通信程序也应能检测到错误或错误，以确保通信可靠。

通过网络读写 PLC 数据与状态，还必须有权限要求，以确保 PLC 工作及数据安全。

1. 通信可靠：

为保证通信可靠，可用数据校验或重复通信。

（1）数据校验。为确保传送的数据传送准确无误，常在传送过程中进行相应的校验，以便及时发现问题，避免不正确数据被误用。常用的校验有横向与纵向两种：横向，对一个字符的 ASCII 作检测，也就是奇偶校验；纵向，对一串字符作校验。

（2）重复通信。重复通信相当于，人们在谈话时，多问多听几次，听清了再作处理。PLC 通信也类似，一般为两次过程——写数据、要得到“写成功”的应答。读数据、看得到的数据是否正确，及校验码对否。

如 OMRON 公司的 Host link 网，计算机与 PLC 通信时，PLC 总有应答信号送计算机。如应答码为“00”，说明 PLC 已正确执行了计算机的命令，否则为出错。

三菱公司 PLC 三次过程——发读命令给对方、从对方取得数据、再向对方已取得数据

的应答。

为了确保通信正确，有时使用更为可靠的办法。这些办法是：

多次通信传送同一数据，接收方用“表决”的方式确定所收到的数据。如发送方对某数据发送三次，接收方收到的为 01011001，01011001 及 01001001，由于 01011001 收到两次，故确认收到的字符为 01011001。这当然也是冗余，是帧发送冗余。

再有，也可是接收方收到数据后，再把相同的数据回传给发送方，发送方再作检查。这也叫回声检测（echochecking）。发送的与回传的相同，则通信无误。如不同，说明有误，进而再作相应处理。

2. 通信安全

通信安全主要指，在通信中数据的读写及节点间互操作，要有权限设定。不同的人有不同的操作权限。如上位计算机对 PLC 的操作，有的人权限最高，可读写 PLC 数据，可操作 PLC。而有的人只能读写。有的只能读，不能写。有的什么操作都不允许。

有了权限设定，通信安全将有保证。

8.6 PLC 异常处理程序

关键词：预测错误、安全控制、错误诊断、误动作避免、掉电保护、应急处理、报警系统

正如用 VB、VC、DELPHI 等语言编写的计算机程序，都有对异常情况的处理一样，PLC 也应有异常处理程序，用以 PLC 的种种异常情况及出错处理。具体处理的项目如：掉电保护、错误预测、错误预防、错误报警、错误记录、错误控制及错误诊断。

目前，对于计算机程序，有人提出提高强壮性（即鲁棒性）问题，即每一个函数都尽可能地保证是独立完备的、安全的；程序基本完成后，从最底层的类开始，逐步找出每个对外接口的前条件（前提）和后条件（结果），然后，判断有无可能发生错误，并且决定这些错误应该如何处理。而在处理错误上，应预防所有可以预料和防止的错误；处理所有可以预料但不能防止的错误；捕获所有不能预料的错误。

这些，对 PLC 程序的可靠性设计及异常处理程序设计也是很有意义的。

1. 掉电保护

控制对象工作过程中，有时出现电源突然掉电，过后又恢复，这是常见的异常现象。对此要区别对待：

（1）电源恢复后不继续工作。要求工作人员对系统作初始化、重起动，才能重新工作。这样的程序必须设计成电源掉电而又恢复时，不能使各工作部件工作。实现它的办法是：各个动作加自保持（一旦失电，不起动不能再得电）及做必要的联锁。

（2）电源恢复后要继续工作，继续原来的顺序进行。这样，最好于掉电时能记录下掉电前的情况，当电源恢复后，对象仍可自动的按原顺序继续工作。这时就用到掉电保持的器件，如用保持继电器代替内部继电器，用计数器代替定时器。所设计程序也要考虑到前后衔接。

图 8-18 就是图 4-31 的改进。目的是能处理掉电问题。

从图 8-18 知，它把图中的 X1、X2、X3、M 代之以保持继电器 HR0.01、HR0.02、

HR0. 03、HR0. 04。Z（内部继电器）的物理地址指定为保持继电器 HR0. 00（图未示出）。接着，再用 HR0. 01、HR0. 02、HR0. 03、HR0. 04 对应地去控制输出继电器 X1、X2、X3、M。此外，定时器 TM、TL 要改用计数器，并用时间脉冲实现计数。因为定时器掉电是不保持的。

使用该图程序，掉电后再得电，系统将在原来的基础上继续工作。

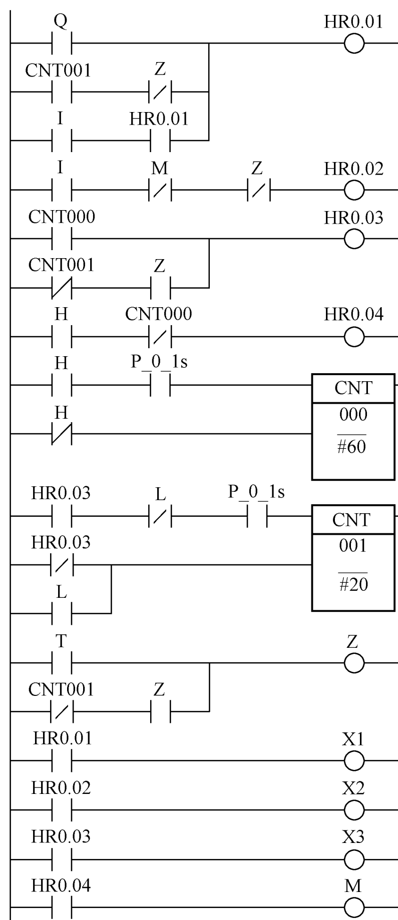


图 8-18　掉电保护程序

2. 出错处理

PLC 有很多标志位，将在指令不同执行时，有不同的取值。如 P_ER，出错标志，程序出错时，此位 ON，再如 P_CY，进位标志，加运算进位时，ON。PID 执行了，它也 ON。

往往可使用这些标志位监视指令是否正确执行，并根据监视情况作作异常处理。

3. 错误报警

在有的 PLC 控制系统中，使用了 3 级错误报警系统。1 级设置在控制现场各控制柜面板，用指示灯指示设备正常运行和错误情况，当设备正常运行时对应指示灯亮，当该设备运行有错误时指示灯闪烁。2 级错误显示设置在中心控制室大屏幕监视器上，当设备出现错误时，有文字显示错误类型，工艺流程图上对应的设备闪烁，历史事件表中将记录该错误。3 级错误显示设置在中心控制室信号箱内，当设备出现错误时，信号箱将用声、光报警方式提示工作人员，及时处理错误。

在处理错误时，又将错误进行分类，有些错误是要求系统停止运行的，但有些错误对系统工作影响不大，系统可带错误运行，错误可在运行中排除，这样就大大减少整个系统停止运行时间，提高系统可靠性运行水平。

当然，为了信息显示的准确，这些设备或指示灯必须保持完好无损。因此应有相应的检查与测试程序。

在故障或出错报警的同时，做好故障纪录，也是必要的。这也可与状态记录一起编程。

4. 错误控制

一旦系统出错，除了报警、记录，马上要考虑的是对出错或故障性质、严重程度的判断，一旦确认是严重故障，应有应急处理机制或程序。能控制住故障，以确保设备安全，特别是人身安全。

一般而言，可将与机器有关的危险隔离，主动或被动地将它封住，或者在探测到危险时即时终止过程以免人员受伤等。

另外，隔离危险，防止接近危险，或者在探测到危险时即时终止过程，这是惟一能把握

并能避免死亡或受伤、同时优化生产过程的机会。

这里最简单方法是设备紧急停车。或使 PLC 禁止输出等。

总之，应在程序中考虑这些措施，确保故障能得以控制。

5. 状态记录

飞机失事后的第一件事就是想方设法先找到黑匣，因为它记录着飞机的飞行数据，根据它很容易查找、判断出事的原因。PLC 运行也可有自己的“黑匣”，那就是 PLC 的数据区。而且，现在这数据区已相当大。只要编有相应的 PLC 运行情况数据记录，就可把它存储在这个数据区中。注意，这里讲的状态不仅是故障，还可是系统运行负荷情况，以及在不同负荷下运行时间，系统的重要性能特性等。一旦 PLC 控制系统出现故障，可找出这个记录，分析这个记录。这对故障判断、定位都将有很大的帮助。

6. 故障预测与预防

设备修理，最原始的方法是，坏了修，不坏不修。但重要设备长期不修的话，一旦突然坏了，给生产带来的损失将是很大的。

为此，用了计划预修，即使用时间长了，坏不坏，都强迫修理。这可减少突然坏了给生产带来的损失。但资源却不能得到充分利用。

最好的办法是故障预测与预防。用传感器不断监测设备的工作状态参数，并记入 PLC 的数据区。再由 PLC 实时判断，可视情况，对可能的故障进行预测，或提示维护，或提示停机修理，以作必要预防。对机械设备，一般检查轴承噪声及滑油变脏的时间。一般讲，噪声变大、滑油变脏时间缩短，将是需要维修的征兆等。

事实上，只要做了有关配置，用 PLC 程序完全有可能实现这种故障预测及预防。这样既可充分利用资源，又不会因设备突然损坏给生产带来损失。

7. 故障或错误诊断

故障或错误诊断是对已出故障或错误的定性与定位，为排除故障、纠正错误提供依据。为此，可在计算机上，建立故障或错误诊断知识库，运行系统监视与诊断程序。PLC 在现场监视系统工作，实时监测系统状态，采集与存储有关数据。必要时，两者联机、通信，PLC 把采集及存储的有关数据的传送给计算机，计算机处理这些数据，并存入数据库。一旦系统出现故障，知识库即可根据知识库的规则及推理机制，对故障进行实时诊断。每次诊断后，知识库的学习机制还可丰富、修改知识库的有关规则，使知识库的功能不断为增强。当然，在这样知识库的建立与使用上，PLC 是无能为力的。但 PLC 所提供实时数据，则是计算机进行诊断的基础与依据。

1992～1996 年，组建北京航空航天 FMS。该 FMS 由三台加工中心，刀流、物流、有关辅助系统及操作员站、工程师站、故障诊断站组成。用于自动化加工大型飞机零件，是当时国家科技攻关的重点课题。系统庞大且复杂，一旦出现故障，故障诊断是个大难题。而若故障不能及时诊断，将增长排故时间，进而影响系统的工作效率。

为此，国家国防科工委责成北京航空航天大学洪士勤教授领导的“北京航空航天 FMS 实验中心故障诊断知识库”课题组，对该故障诊断站进行研究。他们先后用 PROLOG 及 BORDLAND C＋＋31 编程语言，为该诊断编写了基于故障树组织的故障诊断知识库，并用 OMRON C60H 型 PLC 进行系统工作状态在线监测与关键数据实时采集。实践证明，有了 PLC 与计算机联机提供的这些实时数据，不仅可减少运行该工作站知识库时大量的手工操

作，而且，还可进行故障的在线诊断，大大提高故障诊断的及时性及准确性。在此，PLC 是功不可没的。

8. 冗余配置与程序设计

在重要的使用场合，PLC 冗余配置已用得越来越多。以至于还有用容错系统。冗余有电源冗余、CPU 冗余、I/O 冗余及网络冗余。

冗余主要用硬件处理。软件上，如 OMRON 公司用 CPU 冗余，其实所编的程序与非冗余是一样的。虽有两个 CPU，但运行的程序完全相同。出现故障后 CPU 间的工作切换是系统自动完成的，无需人工干预。

但有的厂家 PLC 也有冗余软件配置的。那样，也可能也要编写少量的用户程序。

结语

PLC 用于种种控制，可实现系统自动化，用于联网、通信及信息处理可实现信息化。但也因此带来大系统的出现及系统的复杂化。这样，如果个别部分出现故障，将相互影响，其结果可能是灾难性的。为此，防止故障扩大及进行系统可靠性设计是绝对必要的。

如果进一步发展，最好能实现系统控制冗余及智能化。出现故障仍可工作，同时还能自己诊断，使系统能很快恢复正常。

可靠性当然不只是软件的问题，在硬件上采取措施更为重要。在硬件上的措施，PLC 厂商多有建议，应按相应规定处理，不可马虎。

请想想

1. PLC 控制可靠性含义？
2. PLC 控制可靠性意义？
3. PLC 提高自身可靠性有哪些措施？
4. 提高 PLC 输入可靠性可采取的措施？
5. 提高 PLC 输出可靠性可采取的措施？
6. 程序异常处理的方法？

请试试

1. 编写联锁控制程序
2. 编写互锁控制程序
3. 设计一种滤波程序
4. 其它实际控制程序设计

第 9 章　PLC 程序组织

9.1　PLC 程序组织的重要性及方法

关键词：程序组织、工程、硬件配置、符号编辑、初始化程序

以前各章已对 PLC 实现各种功能的编程作了讨论。本章将讨论怎样把各个功能不同的程序合成一个整体，进而组成一个工程，即 PLC 程序的组织问题。

9.1.1　PLC 程序组织概念

程序组织是指，怎样把各个功能不同的程序块，再加上硬件配置、相应的设定及地址分配，组成一个工程的过程。所以，PLC 程序组织的目的，就是组织工程。一般讲，一个 PLC 的工程要包含一个或多个 PLC 的硬件配置、地址分配、有关设定及一系列程序或程序段。具体的一个工程的内含，与 PLC 品牌、类型、型号及所用的编程软件有关。

工程还是 PLC 程序、设定及其它数据的存储单位。OMRON PLC 存储工程文件的扩展名为 CXP。它是压缩后的文件，是二进制码，不可视的。还有扩展名为 CXT 的工程文件，是文本文件或图表文本文件，未经压缩，可用文本阅读器阅读。

CXT 的工程文件与 CXP 的工程文件可相互转换。如用 CXP 软件调出 CXP 工程文件，把它存成扩展名为 CXT 工程文件，即实现了前者到后者的转换。反之，也一样。OMRON 公司老的编程软件编的程序，也可转换为 CXT，进而转换为或直接转换 CXP 的工程文件。

有的公司 PLC 的工程要用多个文件存储，并分布在多个子文件夹中，各个文件存储不同类型的数据。

图 9-1 所示为用 OMRON CXP 软件，组织的一个 OMRON PLC 的工程。

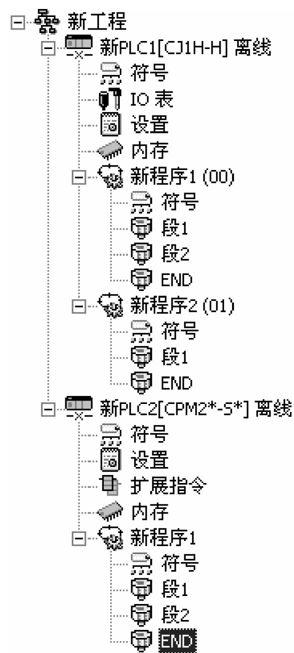


图 9-1　工程实例

从图 9-1 知，这个工程有两个 PLC，一为 CJ1H-H 型，另一为 CPM2 型。CJ1H-H 型 PLC 有 2 个程序，其中的新程序 1 有 3 个程序段，即，段 1、段 2 及 END 段。CPM2 型 PLC 只有也只能有 1 个程序（它不能多任务编程），该程序也有 3 个程序段，段 1、段 2 及 END 段。这里的各程序段就是以前各章讨论的各种程序。

从图 9-1 还可知，除了程序合成，还有 PLC 型号选定，如这里选 CJ1H-H 型 PLC、CPM2 型 PLC。此外，还有符号表设计、I/O 表设计（CPM2A 不是模块型 PLC，无此设计）及进行有关设置。

显然，只有进行了程序组织，形成一个 PLC 工程，并对所编程序进行编译，且与 PLC

联机（本例有 2 个 PLC，须分别与 2 个 PLC 联机），再把工程中的程序、设置等全部下载给 PLC，PLC 才能正确运行这个工程中的程序。

组织工程是编程最开始的工作。没有建立工程，也没有编程环境，故无法编程。建立工程是编程的最后工作，只有通过工程进行程序编译，与 PLC 联机，并下载程序、设定等给 PLC，所编的程序才能起作用。工程还是 PLC 编程及设定等数据存储的容器，没有工程，这些数据无法保存，也无法调用。

提示：如使用简易编程器编程，其所有操作（包括种种设定）全用手工实现。所以，也谈不上什么工程。

9.1.2　PLC 程序组织任务

PLC 程序组织的任务有：硬件配置、I/O 分配、符号表设计、设置、初始化程序设计及有关程序合成。

1. 硬件配置

选定 PLC 的机型、CPU 型号；

使用网络及网络通信参数；

机架及模块选定及编址；

符号编辑与内部器件地址对应。

所有这些，在组织程序时，都必须在软件的有关窗口中选定。具体见第 2 章的图 2-90。

2. I/O 设计

对模块式 PLC，在程序组织时，才有 I/O 表设计问题。当要自行设计时，可双击工程工作区中的“I/O 表”项。击后，将弹出 I/O 表设计窗口，如图 9-2 所示。

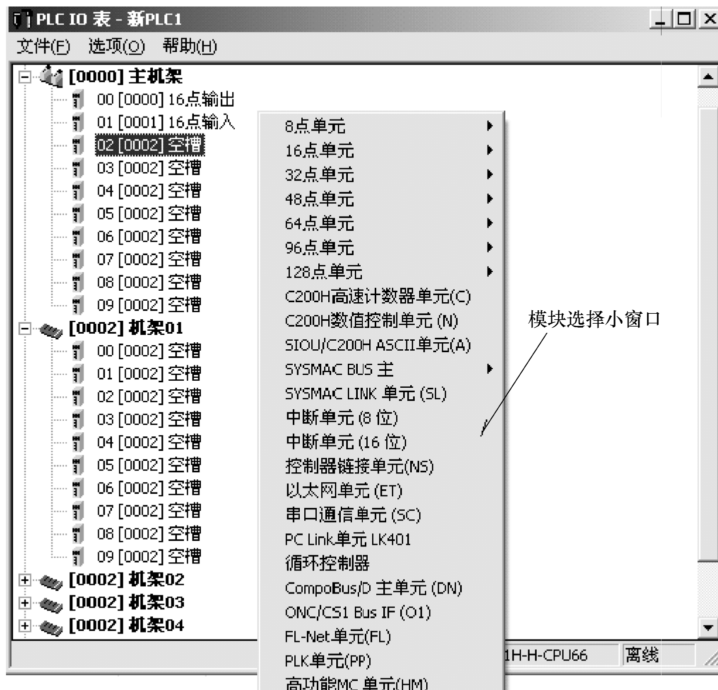


图 9-2　I/O 表设计窗口

从图 9-2 知，利用该窗口，可对每一机架、槽位选用什么模块以及模块的有关参数进行选择。提供了可能的 I/O 配置。可按你的系统实际配置进行选择。该图正对机架 01 的槽位 02 进行配置。这是选定槽位 02 后，再右击鼠标弹出的选择小窗口，从中可选择有关的模块。

I/O 表设计以及登记的细节，已在本书第 2 章介绍 CXP 软件作了说明，这里不再重复。

3. 符号表编辑

在进行程序组织时必须编辑符号表，务实所有的符号都有确定的实际地址与其对应。表 9-1 所示的为 CP1H 型 PLC 系统提供的全局符号变量。

表 9-1　CP1H 机系统提供的全局符号变量

名　　称	数据类型	地址/值	使用	注　　释
· P_0_02s	BOOL	CF103	工作	0.02s 时钟脉冲位
· P_0_1s	BOOL	CF100	工作	0.1s 时钟脉冲位
· P_0_2s	BOOL	CF101	工作	0.2s 时钟脉冲位
· P_1min	BOOL	CF104	工作	1min 时钟脉冲位
· P_1s	BOOL	CF102	工作	1.0s 时钟脉冲位
· P_AER	BOOL	CF011	工作	访问错误标志
– P_CIO	WORD	A450	工作	CIO 区参数
· P_CY	BOOL	CF004	工作	进位(CY)标志
· P_Cycle_Time_Error	BOOL	A401.08	工作	循环时间错误标志
= P_Cycle_Time_Value	UDINT	A264	工作	当前扫描时间
– P_DM	WORD	A460	工作	DM 区参数
– P_EM0	WORD	A461	工作	EM0 区参数
– P_EM1	WORD	A462	工作	EM1 区参数
– P_EM2	WORD	A463	工作	EM2 区参数
– P_EM3	WORD	A464	工作	EM3 区参数
– P_EM4	WORD	A465	工作	EM4 区参数
– P_EM5	WORD	A466	工作	EM5 区参数
– P_EM6	WORD	A467	工作	EM6 区参数
– P_EM7	WORD	A468	工作	EM7 区参数
– P_EM8	WORD	A469	工作	EM8 区参数
– P_EM9	WORD	A470	工作	EM9 区参数
– P_EMA	WORD	A471	工作	EMA 区参数
– P_EMB	WORD	A472	工作	EMB 区参数
– P_EMC	WORD	A473	工作	EMC 区参数
· P_EQ	BOOL	CF006	工作	等于(EQ)标志
· P_ER	BOOL	CF003	工作	指令执行错误(ER)标志
· P_First_Cycle	BOOL	A200.11	工作	第一次循环标志
· P_First_Cycle_Task	BOOL	A200.15	工作	第一次任务执行标志

（续）

名　　称	数据类型	地址/值	使用	注　　释
· P_GE	BOOL	CF000	工作	大于或等于(GE)标志
· P_GT	BOOL	CF005	工作	大于(GT)标志
– P_HR	WORD	A452	工作	HR 区参数
· P_IO_Verify_Error	BOOL	A402.09	工作	I/O 确认错误标志
· P_LE	BOOL	CF002	工作	小于等于(LE)标志
· P_Low_Battery	BOOL	A402.04	工作	电池电量低标志
· P_LT	BOOL	CF007	工作	小于(LT)标志
= P_Max_Cycle_Time	UDINT	A262	工作	最大循环次数
· P_N	BOOL	CF008	工作	负数(N)标志
· P_NE	BOOL	CF001	工作	不等于标志(NE)
· P_OF	BOOL	CF009	工作	上溢出(OF)标志
· P_Off	BOOL	CF114	工作	常断标志
· P_On	BOOL	CF113	工作	常通标志
· P_Output_Off_Bit	BOOL	A500.15	工作	输出关闭位
· P_UF	BOOL	CF010	工作	下溢出(UF)标志
– P_WR	WORD	A451	工作	WR 区参数
· P_步	BOOL	A200.12	工作	步标志

该表列的符号是系统提供的全局变量符号。用户程序也可定义自己的全局符号变量。要通过编辑指定其名称、类型、地址。如需要，也可加注释。

全局变量在整个 PLC 中有效。也可定义局部变量，它仅在所定义的程序中有效。如在新程序 1 中定义的变量，仅在新程序 1 中有效，在新程序 2 中无效。

4. 设置

设置是指，在程序组织时所进行的种种项目设置，以确保 PLC 能正常工作。

从前这些设定，都是直接在 DM 设定区中写有关数据。而今有了 CXP 软件，可在设置窗口及弹出的窗口上进行。而这个设置的内容，又因机型不同而有所不同。图 9-3 所示为 CP1H 型 PLC 的设置窗口。图 9-4 所示的则为 CJ1H-H 型 PLC 的设置窗口。

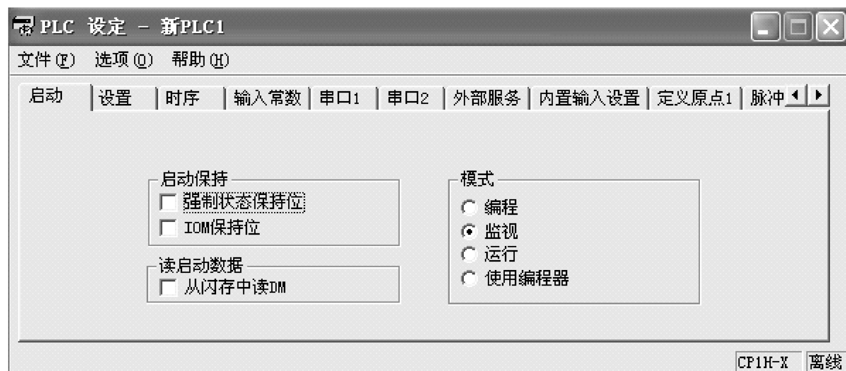


图 9-3　CP1H 型 PLC 设置窗口

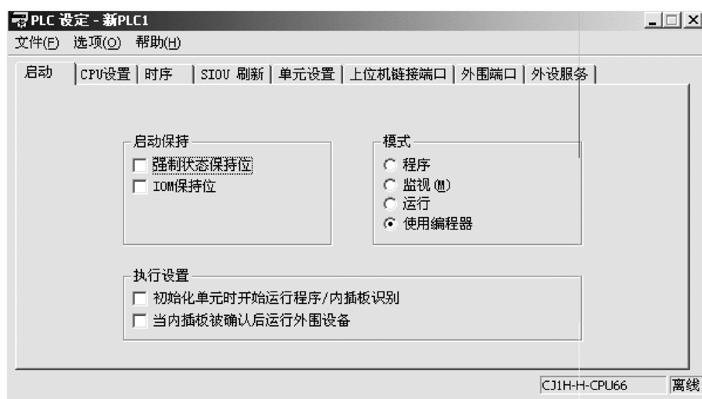


图 9-4　CJ1H-H 型 PLC 设置窗口

从这两张图知，这里各有各的设置项目。CP1H 型 PLC 设置的项目有启动、设置、时序、输入常数、串行接口 1、串行接口 2、外部服务、内置输入设置、定义原点 1、脉冲输出 1、脉冲输出 2、脉冲输出 3、SIOU 刷新、FINS 保护等。而 CJ1H-H 型 PLC 设置的项目有启动、CPU 设置、时序、SIOU 刷新、单元设置、上位链接端口、外围端口、外围服务等。这些不同反映了 PLC 功能与性能的差异，可参阅有关说明书，作相应选择。

5. 初始化程序

PLC 上电时，系统会对自身进行初始化。如定时器复位，除 DM、HR 及计数器外的全部内部器件清零，以及种种设定的读入及生效等等。有的 PLC，如 CQM1，若软设定时 DIP 开关针 2 设成 ON，则在 PLC 上电时会自动把内存卡中的程序装载到 CPU 的内存中，等等。故多数 PLC 自身的初始化问题，用户可不必理会，了解清楚怎么回事就可以了。

而初始化程序是指，用 PLC 运行时，仅在第 1 扫描周期运行的程序，进行用户所希望的初始化。其目的主要是，实现用上述方法设置无法进行的设置及有关初始数据的赋值。

初始化程序一般通过 25315 特殊继电器调用。它在 PLC 进入运行（或监控状态）时，ON 一个扫描周期。初始化程序是用户自己要编的。

若要对步进程序（使用 STEP 指令）作初始化，还可使用 25407 特殊继电器，它在进入 STEP 新步，ON 一个扫描周期。

初始化程序有两个方面：

（1）数据初始化主要是对要使用的数据赋初值。如 100 通道要使用数据，有两种办法处理：一是用立即数写入，如图 9-5a 所示；另一是用存于 DM、HR 中（它为掉电保持），然后再传给 100。如图 9-5b 所示。

从图知，这里先进行 DM0 与立即数 0 比较，若相等，则把即时数 50 传 DM0，然后，再把 DM0 传给 100 通道。若 DM0 不为 0，则把 DM0 当时的内容传给 100 通道。这样可以做到程序初次执行时，可按即时数（这里的 50）设定，若不合适进行了修改，将按修改后的数设定。

（2）设定初始化有的特殊单元在上电工作时，要运行一些初始化程序，才能进行工作，如温度显示单元。图 9-6 所示即为温度显示单元初始化程序。这里假设温度单元为 TS001，为热电偶单元，机号设为 00，检测用热电偶为 K（CA），温度范围为 0～800℃，设为 07。

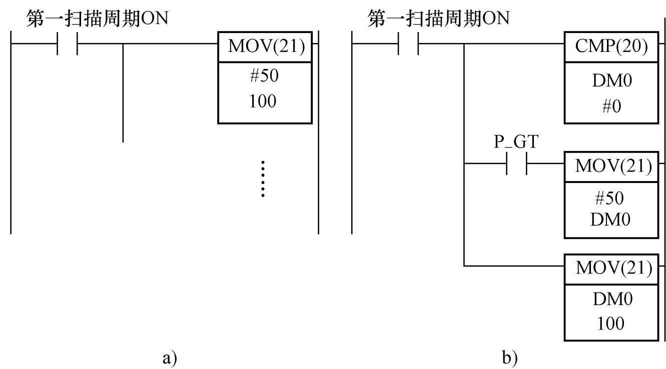


图 9-5　数据初始化程序

图中 105. 06 为 TS 单元等待设置标志位，在设定温度范围过程，它保持 OFF，直到设定完成。完成后，若设定无误，即 105. 00 OFF，则将使 100. 15 ON，使所作的设定生效。若 105. 00 ON，说明设定不当，将报警，可提醒重新设。

图 9-7 所示的为 CPM1A-MA002 初始化设定程序。运行它后，可实现输入 4 路，量程 1 ~5V,输出 1 路，量程 0 ~10V。不用求平均数功能。

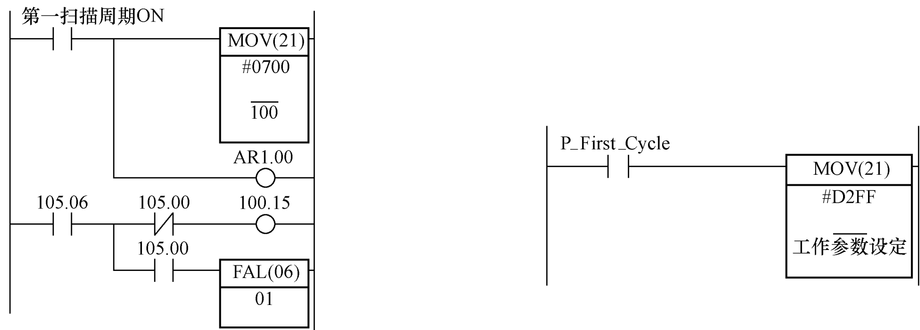


图 9-6　TS 温度单元初始化程序

图 9-7　CPM1A-MA002 初始化设定程序

再者，定时中断、高速计数启用比较指令等，也都要有相应的初始化程序。

可知，这些设定初始化程序是必须的，无此程序，PLC 的预期工作或功能将无法实现。

提示：初始化程序中赋值的数据，不应被后执行的指令修改，如图 9-7 中的“工作参数设定”通道的内容，在第 1 扫描周期，绝对不应被执行别的所修改。 否则，这个设定程序将失效。

6. 程序合成

程序合成看起来简单，好像只是把各部分的程序集成就是了。其实，这里并不简单。关键是这里的“各部分”怎么确定的。要先分，然后才有合。这就是将要讨论的程序组织方法。

PLC 程序组织方法与 PLC 的品牌、类型、机型有关。大体上有，模块化组织、多任务组织及面向对象编程。

7. 多工作模式

电路状态的设置，控制电路应可处在不同的工作状态，以进行不同的控制。常见的有：

（1）手动工作状态：可对对象进行手动操作，以实现工作要求。

（2）调整状态：可使设备作空运转，以实现对设备的调整。

（3）自动工作状态：可以有若干种自动工作状态（工作方式）。

这些状态（方式）可选择一个基本的进行设计，然后再考虑其它状态（方式）。其间靠切换开关实现。

从实用考虑，多数控制电路都是多种状态、多种方式的。故考虑这个问题也是设计电路的重要工作。

9.2 模块化程序组织

关键词：模块化、子程序法模块化、跳转指令模块化、步进指令模块化

9.2.1 程序模块化组织概念

OMRON C系列PLC，还有其它一些厂商的PLC，用户程序是存储在一个统一的存储区中。程序的传送也是一次性的。但人为地可将其分成若干块，以块为单位设计及调试。然后再用主程序，按需要去调用这些块。即人为的先分解程序，后再合成程序，则是可能的。这也就是本节讲的程序模块化组织。

程序分成模块，或程序模块化组织的优点是：

（1）程序较清晰，可读性强；

（2）程序便于更改，也便于扩充或删节，可改性好；

（3）程序可标准化，特别是一些功能程序，如实现PID算法的程序，可编成标准的。事实上西门子公司就提供不少标准程序。用户移植使用这些标准程序可大大简化编程。

（4）程序设计与调试可分块进行，把难点分散，便于成功。块小、变量少，也便于用多种逻辑设计的方法设计程序。

（5）程序模块化还可实现多人参与编程，提高编程的速度。

（6）在存在重复调用一种模块的情况下，可不必重复编写要调用的模块的程序，可减少程序量。

（7）在存在不需经常对其扫描的程序块时，还可节省扫描周期，提高PLC的响应速度。

程序模块化组织与PLC的指令系统有关。常用的有3种方法：子程序及其调用、跳转及其条件设置和使用步进指令。

9.2.2 使用段、子程序或功能块模块化

段（Section）是编程软件提供的程序组织单位。在程序中，指令的执行过程一般是从最接近程序根节点的段开始执行，然后依次向下，直到离节点最远的段。根据功能把程序分成若干段，分别有指定意义明确的名称，然后再加注解，这样分段编程是最简便的模块化。显然，这样的程序很便于阅读及修改，是编程的好习惯。

程序模块化组织也可使用子程序或功能块。子程序、功能块也按功能划分。特别对多次调用子程序或功能块，不仅可使程序模块化，从而简化程序、减少程序容量，还可实现程序共享。也是编程的好习惯。

9.2.3　使用跳转指令模块化

跳转指令也可用以实现程序模块化。JMP 与 JME 之间的程序组成块，执行 JMP 的条件 ON 后作为对其的调用。由于 JMP—JME 指令可以嵌套，用它也可实现在模块之下的再分块，类似于子程序再调子程序。与子程序相比这里不同的是：多次实现的功能要多次编写，无法编成标准的子程序，多次作调用。另外，程序虽由 JMP—JME 划分块，但毕竟都还是连续地放在一起，阅读是不便的。

9.2.4　使用步进指令模块化

STEP 指令中的每一步的程序也是自成体系的，也可看成一个程序模块。如果整个程序或程序的一部分是依此顺序、分支或平行执行的，则可通过步进指令进行程序的模块化组织。

必须再指出的是，用上述方法把程序划块只是人为的，不像将要讨论的任务划分那样真的划分成模块。故只能称之为模块化。但这种连成一片又划分为模块的方法比真分成模块要直接、简便，熟练地掌握它，也可达到目的。

总之，程序模块化组织的好处是：不易出错，出了错也好找；便于阅读、易于理解和维护。

模块化组织也便于分步设计程序。第一步先划分模块，编出的程序最为抽象；第二步编出的程序是把第一步所编的程序细化，较为抽象；… 第 i 步编出的程序比第 i－1 步抽象级要低；… 直到最后，第 n 步编出的程序即为可执行的程序。程序可从粗到精，一步步推进。

所谓“抽象程序”指，程序所描述的解决问题的处理规则，是由那些“做什么”操作组成，而不涉及这些操作“怎样做”以及解决问题的对象具有什么结构，不涉及构造的每个局部细节。

这一方法原理就是：对一个问题（或任务），程序人员应立足于全局，考虑如何解决这一问题的总体关系，而不涉及局部细节。在确保全局的正确性之后，再分别对每一局部进行考虑，每个局部又将是一个问题或任务，因而这一方法是自顶而下的，同时也是逐步求精的。

9.3　多任务程序组织

关键词：任务、循环任务、中断任务

多任务编程是 OMRON 公司 CS1 型 PLC 推出的编程方法。它可按要求把程序分成多个不同功能及不同工作方式的任务。用任务的种种调用，推进整个程序的执行，达到运行程序的目的。

9.3.1　OMRON PLC 任务划分

任务划分依 PLC 的型号不同而有所不同。大体上任务有两类：循环任务及中断任务。

循环任务大体有 32 个（机型不同，可能有所不同）。任务编号从 0～31。循环任务 00 为起始循环任务，是首先要执行的任务，也是默认要用的任务。如仅一个任务（最少也要

有一个）则就是它。

如有多个循环任务，在 PLC 运行时，总是先按序号从小到大，依次周而复始不断地执行。如被任务管理指令（见后）控制，则怎么管理，怎么执行。

中断任务较多，多达256个，机型不同，可能也有所不同。编号也是从0开始，0～255。

中断任务由各种中断事件触发。有了中断事件，就暂时停止循环任务的执行，转去调用相应中断号的中断任务。而且，发生一次事件，仅调用一次。如果，同时有两个中断事件发生，则先调中断号小（优先级高）的任务，执行小的后，再执行大的。都执行完中断任务，再转回执行循环任务。

图9-8所示为多任务程序执行的情况。

中断任务有：电源断中断任务、定时中断、I/O 中断任务、外中断任务。此外，有的机型还有扩充循环任务，是按循环任务处理的中断任务。编号在8000～8255之间的十进制数（值8000～8255定义0～255扩充循环任务）。

电源断中断任务优先级最高，用中断0号。中断1号、中断2号，用于内部定时中断。其定时间隔可用 CXP 编程软件设定。

I/O 中断要用到中断输入单元。其中断号与中断单元的输入点的编号相对应。如输入点为0，则设定其中断任务号为100，其余类推。

多任务编程是模块化编程的进一步发展。其好处与模块化组织还要多。如图9-9所示，这里任务作了不同的组织，就构成了任务 ABC 及任务 ABD 两个不同的程序，很灵活。

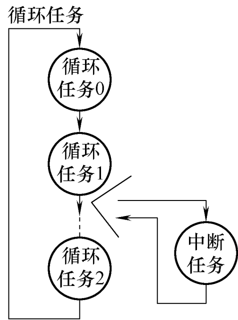


图9-8　多任务程序执行

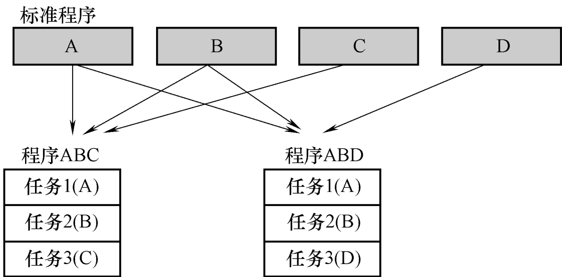


图9-9　任务的不同组织构成不同的程序

用多任务编程时，每一任务的最后一个指令应是 END。它代表任务的结束。END 指令之后的指令不执行。

执行每一任务开始时，所有的标志位，如"大于"、"等于"……均复位为0。每个任务可以有自己的子程序。而且，别的任务不能调用。

但可以设计全局的子程序。这时，所有的任务均可调用。图9-10所示即为全局子程序使用的情况。

从图知，使用全局子程序可减少程序代码。

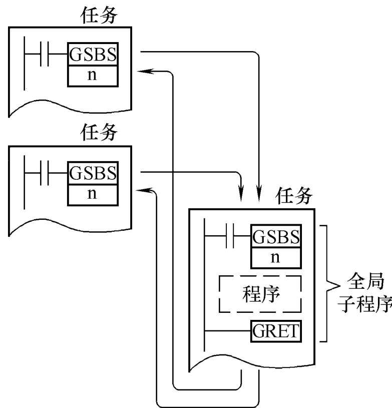


图9-10　为全局子程序使用

9.3.2　OMRON PLC 任务管理

任务管理，也就是任务调用及调用取消。有两种方

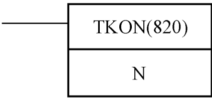
法：由中断事件调用，针对中断任务（无需取消）；用指令调用与调用取消，针对循环任务。

系统提供的指令调用指令为“任务 ON”。调用取消指令为“任务 OFF”。

1. 任务 ON

助记符号为 TKON（820），目的是使得指定的任务执行。

梯形图符号为：



这里，N——循环任务号。

N 应在其允许范围内根据任务的类型指定。必须是十进制 00～31（十进制）之间的一个常数。（数值 0～31 定义任务 0～31）。

对 CS1-H，CJ1-H，和 CJ1M 型 PLC 的 CPU 单元，也可为扩充循环任务号。N 必须是一个在 8000～8255（十进制）间的常数。（值 8000～8255 定义扩充循环任务 0～255）。

执行本指令，可使指定的任务置于可执行状态。并把相应的任务标志（TK00～TK31）置 ON。

如果本指令指定任务号小于本任务号，指定的任务将从下一个循环开始执行。如果指定的任务号大于本任务号，该任务在当前的循环执行。图 9-11 所示为以上两种调用情况的图解说明。

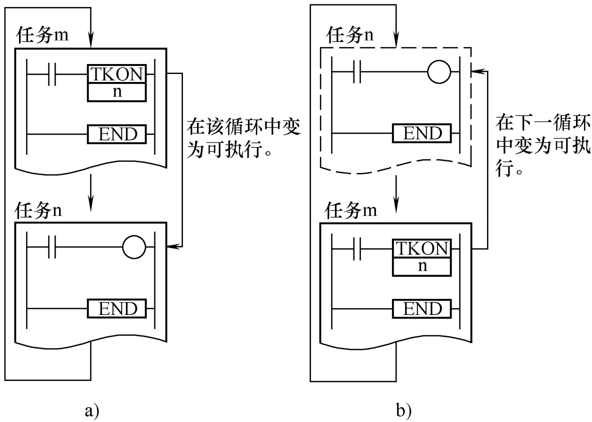
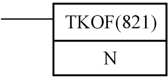


图 9-11　两种调用情况图解
a）指定的任务号高于本任务号（m＜n）　b）指定的任务号低于本任务号（m＞n）

2. 任务 OFF

助记符号为 TKOF（821），目的是把指定的任务置为待机状态，即禁止任务的执行。梯形图符号为：



这里，N——循环任务号。

N 应在其允许范围内根据任务的类型指定。必须是十进制 00～31（十进制）之间的一个常数。（数值 0～31 定义任务 0～31）。

对于 CS1-H，CJ1-H，和 CJ1M 型 PLC 的 CPU 单元，也可为扩充循环任务号。N 必须是一个在 8000～8255（十进制）间的常数。（值 8000～8255 定义扩充循环任务 0～255）。

执行本指令，可使指定的任务置于待机状态，并把相应的任务标志（TK00～TK31）置 OFF。

如果本指令指定任务号小于本任务号，指定的任务将从下一个循环开始待机。如果指定的任务号大于本任务号，该任务在当前的循环待机。情况如同图 9-11 对 TKON 指令的说明。

9.3.3 OMRON PLC 任务组织

任务组织含任务的建立及调用。

中断任务的建立主要是，作好有关软、硬件设定及编写中断处理程序。而它的调用不须组织，由中断事件调用。中断处理程序可按照控制或数据处理的要求编写，与以前讨论的没有本质差别。这里不赘述。

循环任务除任务 00 外，所有的都要另行建立。调用也要组织。

建立循环任务时，应在工作区中先选择 PLC 项，再选择插入“新程序”。然后单击程序的属性项，将弹出程序属性窗口，如图 9-12 所示。

从图 9-12 知，可从中选定本程序的任务类型及任务编号。同时，还可选定，操作开始时是否执行本任务。如图 9-13 所示为循环任务 02 的操作开始选择画面，其名称为“新程序3”，而且，操作开始时执行。

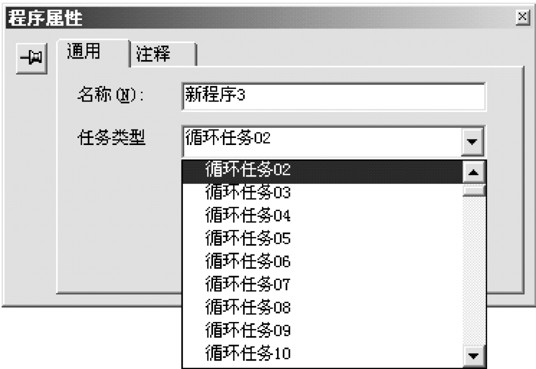


图 9-12 程序属性窗口



图 9-13 循环任务 02 的操作开始选择

任务除了一开始就让它执行，也可开始时不执行，而在某个条件下才执行。图 9-14 所示为这样的例子。

从图知，当 D0 的值增加到#50 时，才执行 TKON 1 指令，才把任务 1 调用。

已调用的任务还可在一定的条件下，让其停止执行。图 9-15 所示为这样的例子。

从图 9-15 知，当 D2 增加到#30 时，将执行 TKOF 1 指令，将使任务 1（本例即自身这个任务）停止执行。

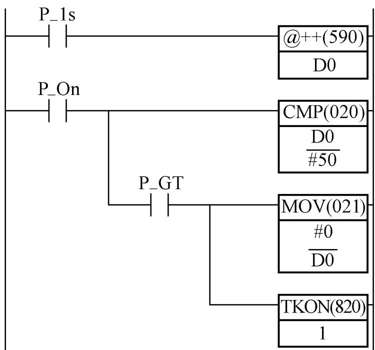


图 9-14 任务调用例子

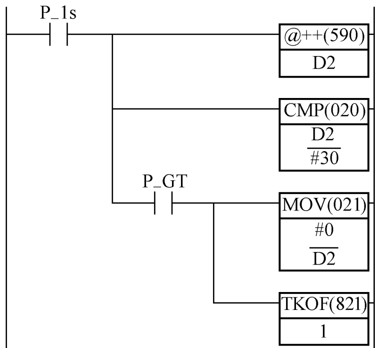


图 9-15 使任务停止执行例子

可知，在多任务编程的情况下，编写好各任务的代码后，可很方便地根据任务是否执行的情况进行组织。

9.4　PLC 程序柔性化

关键词：柔性化、程序柔性、参数柔性、地址柔性、动作柔性、反馈柔性

柔性（Flexible），这里讲的柔性化设计是指，所设计的程序很灵活，可不修改或仅作简单的修改就能应用于新的类似情况。并且即使要修改也很方便。显然，能设计出这样的程序是我们追求的目标。

这里讲的柔性有 5 个方面内容：程序使用柔性、地址分配柔性、参数设定柔性、动作选择柔性及信号反馈柔性。

9.4.1　程序使用柔性

我们可为 PLC 编写很多程序，但在实际工作中，使用哪个程序可以是柔性的。有两个办法：一是全局的，二是局部的。当然，它的实现要有相应的硬件支持。

1. 全局柔性

全局柔性指使用的程序可全局更换，以使 PLC 适应于不同的工作要求。

对微型机，如 OMRON 的 SP10、16、20 型机，它有程序存储卡，可存 26 个程序。可按要求把这 26 个中的一个装入 PLC，使 PLC 能灵活地做相应的工作。

再如，有的 PLC，其程序可固化到 PROM 中，可多编几套，各有各的用途，也可达到柔性的目的。

2. 局部柔性

局部柔性指使用的程序可局部更换，以使 PLC 适应于不同的工作要求。

高性能机型，如 C2000H、CV 型机等，有从存储卡上读入程序的指令，即 FILP。执行它可把存储卡中的某段程序读入到 PLC，并存储于此指令之后的程序存储区中，之后即执行此新的程序。而原来的程序被新的更换，不起作用。

PLC 使用的程序，用这种全局或局部调换达到柔性，是相当方便的。只是它还要相应的硬件条件，还要设计多套或多段程序。

9.4.2　地址分配柔性

早期，PLC 指令的操作数多是 PLC 的实际地址。地址一旦变更，程序就要作相应的改动。尽管，编程软件有修改变量的编辑器帮助修改，但是很麻烦的。而且，还易出错。此外，指令的操作数多是 PLC 的实际地址，没有明确的涵义，程序很难读。

如今，很多 PLC 编程软件可支持用符号地址编程，即 PLC 指令的操作数，用有某种涵义的符号名代替实际地址。用这样编址的程序，读起来就比较好理解了。而且，用符号地址编程，地址可灵活分配，为程序的修改、重用提供很大方便。这也正是地址分配柔性的体现。

用符号地址编程可先编辑符号地址，确定符号地址与实际地址的确定对应关系，然后再用这个符号地址编写程序。也可先按符号编写程序，后再编辑符号地址，确定符号地址与实

际地址的确定对应关系。一般支持符号地址编程的软件都允许这么做。

允许后一种做法，意味着程序地址重新分配时，程序本身可不用改动，只需重新编辑符号地址就可以了。

符号地址还可定义为全局符号及局部符号。全局符号在全局，整个工程中起作用。而局部的符号只在局部，即某程序中起作用。而且，符号名可重复定义，但局部的优先级高，以局部的符号为准。

一般提倡多用局部符号地址。这样，在修改程序时影响面更小些。而且，用局部符号地址，还便于多人参与编程。只要实际地址区分好，各用各的符号，互不影响，很易协调。

9.4.3　参数设定柔性

PLC 程序多少总有一些参数要设定，如定时器的定时值，或比较指令的比较值，等等。这些参数可直接送入常数，用这个常数作设定数。这么做当然是可以的，多数程序也是这么做的。但这样做时，若要改变参数的设定值时，就得改变程序，不大灵活。

其实，如果出现参数需改变的情况，可指定内部器件的相应通道存这些设定值。改变该通道的内容，也就可改变设定值。而改变通道内容可不必动程序，用终端设备或编程器当 PLC 在线工作时都可以改，要灵活与方便得多。

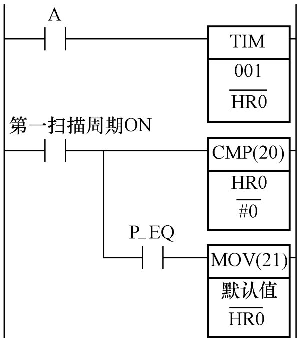


图 9-16　定时器定时值设定

图 9-16 所示的定时器 001，其设定值就存于 HR00 通道中。

从图 9-16 知，HR00 通道内容为零（未对其指定一个值），当 PLC 起动，进入运行状态时，由于 25315 会 ON 一个周期，可把默认值送入 HR00 中。这时，TIM 001 的设定值即为默认值。

若要改变这个值，可通过编程器或终端设备实现，如将其改为#0060，由于 HR 有掉电保持功能，PLC 停止工作，此值可被保留。当程序再启动时，由于 HR00 的值为#0060，不等于 0，25506 为 OFF，将不执行传送指令。故这个改后的#0060 会一直保留。显然，若再改成别的值，情况也完全相同。

设定值也可由输入通道直接确定。这时输入通道接拨码开关，开关的指示值，即可作为设定值。如上例，不用 HR00，而用 000 通道，并把 000 通道的 16 位和拨码开关（一个开关 4 位，共接 4 个开关）的对应点相接。那么，TIM 001 的设定值即可由这个拨码开关设定。

这么处理之后，可使定时控制很灵活。但要使用不少输入点，将增加了硬件开销。为减少硬件开销，也可通过编码传送，实现一字多用，或一数位（digit）多用。图 9-17所示即为一个实例。

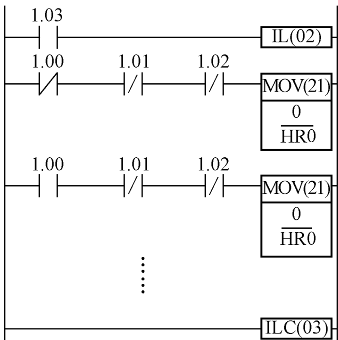


图 9-17　拨码开关设定实例

从图 9-17 知，这里用 001 通道的 00 ~ 03 位作为编码位，000 通道仍然接拨码开关，用以产生设定值。01 通道的 00 ~ 02 位，用以选择地址，3 位二进制数可选 8 个地址。01 通道的 03 位作传送使能位。它 ON，可实现设定值传送。

本例是，当使能位 ON 时，把 000 通道的内容（由拨码开关设定）送给由 01 通道 00 ~ 02 指定的地址通道。本例的地址分别为 HR00 ~ HR07。

本例共用了 20 个输入点，可使 8 个参数由外部拨码开关确定，是较合算的。

有的 PLC，如 OMRON 的 CP1H、西门子的 S7200 型 PLC 机等，以 S7200 型为例，其面板上设有 2 个模拟量输入电位器。电位器旋钮处于不同位置时，可使与其对应某通道（CPM2A 为 249、250 通道）具有不同的值，可在 0 ~ 200 之间作变化。利用好它，也可使参数的设定实现柔性化，还不占用输入点。

9.4.4　动作选择柔性

从关系上讲，PLC 主要用来反映或实现输出与输入间的对应关系的。动作一般由输出产生。如果输出与输入之间的关系用指令直接体现，那动作就没有选择的柔性。

最好是输入与输出间的逻辑关系是间接实现，允许其间有所选择。本书第 2 章第 2.4.6 节讲了输出转换程序（如图 2-74 及图 2-75 所示）。这就是动作选择柔性的设计。

这种动作选择带有柔性的设计，如动作有变化，只需改变动作选择部分的程序，是很简单、较容易的。

这样设计的另一好处是，如要禁止输出，可另加控制开关。如它的输出还有别的，如手动控制，可把所有相关控制接点并联后，作用于它的输出点。

本书第 3.8 节的若干设计举例，都体现了这个动作选择柔性的思想。

9.4.5　信号反馈柔性

输出信号反馈最好也是柔性的。办法也是作些地址转换，把与反馈信号对应的输入先转换到某工作位。再用这个工作位，供程序处理使用。

这样，如要改变反馈作用，程序的逻辑部分可不用改动。而只要把改变输入与某工作位的相应关系就可以了。

这点，在本书第 2.4 节介绍输出转换程序之后，对输入转换的有关问题也作了说明。本书第 3.8 节介绍混合控制中的设计举例，也都体现了这个反馈选择柔性的思想。

总之，柔性化的办法，先是用符号地址编程，用间接数作参数设定；再是程序按功能划分，可分为本体部分、动作选择部分及反馈选择部分。

本体部分用以处理“假想”的输入、输出间的变换，但不直接与输入、输出点关联。真正的关联在动作选择及反馈选择部分。在动作选择部分，把“假想”输出与真正输出关联。在反馈选择部分，把“假想”的输入与真正输入关联。

程序柔性化设计要麻烦一些，但好处是，程序很灵活，稍作以至于不必作改动，即可适应很多不同的情况。

更理想的柔性设计最好能突破按算法设计程序的模式。所设计的程序也许只是框架，具体的算法、实现过程，可在程序运行中形成与完善。这可能是未来要做的事情。

9.5　PLC 面向对象编程

关键词：线性编程、结构化化编程、面向对象认识、对象、类、OOA、OOD、OOP、

组件编程

面向对象编程是结构化编程的发展，是计算机软件技术的一次革命，在软件开发史上具有里程碑的意义。

由于有独特的优点，它已成为计算机编程的主流方法。PLC 编程也逐渐开始运用这个方法。但由于 PLC 程序往往有赖于硬件，而硬件又是千差万别，所以，PLC 编程用面向对象方法难度较大，目前用的还不是太多。然而，用它的编程思想、技巧还是应该的。特别 PLC 厂商开发编程工具软件，不少已用到这个方法。

9.5.1 面向对象编程概念

1. 面向对象认识

对象（Objective），指的是事物，是很广泛的概念。可以说，凡是事物都是对象。如数字 8 是对象，一张桌子也是对象，等等。

“物以类聚”，所有事物总是按一定特征归入一个相应的类（Class）中。因而，可以这么说，类是对象的抽象，而对象则是类的实例。

如上述 8，可归入数学的自然数类。自然数是很多与 8 类似的事物的抽象，而 8 则是自然数这个类的一个实例，也具有自然数的特征。再如上述桌子，可归入日常生活用品的家具类。家具是很多与桌子类似的事物的抽象，而桌子则是家具这个类的一个实例，也具有家具的特征。

在实际生活中，谁见过抽象的“自然数”、抽象的“家具”？没有。见过的，只能是具体的“8”、具体的“桌子”。但停留对具体的“8”、具体的“桌子”的认识，够不够？不够，还得提高到“自然数”、“家具”这样抽象的认识。

对象抽象为类后，将有其特性及处理方法。如 8，代表数量，就是它的一个特性，8 可与其同类的数相加，就是处理它的一个方法。掌握了对象的特性、方法，显然也就容易驾驭它了。

这里的“对象”与“类”反映的是个别（具体）与一般关系，是正确认识事物方法的精髓。正确认识事物，总是先认识“个别”、“具体”，后认识“一般”、“抽象”，再进一步认识新的“个别”、“具体”。从个别到一般、具体到抽象，又从一般到个别、抽象到具体，循环往复，以至无穷。才使得人们对事物的认识不断提高、不断深化。所以，把事物用这种“对象”的方法去分析、理解，也就是说用这样“面向对象”去认识，是很合乎正确的认识事物的方法的。

很显然，这个“类”也是相对的。一个“类”可能还属于外延更大的“另一个类”中。这“另一个类”一般称为“父类”。而这个“类”则是“另一个类”的“子类”。当然，这“父类”之上，也可能还有“父类”。而“子类”之下，也可能还有“子类”，等等，形成一个类的长链。

正如人类父子之间有继承关系一样，在类的长链中，“父类”、“子类”也都有继承关系。“父类”的特征往往在“子类”中有所体现，而对“子类”说，则是对“父类”的特征有所继承。

应强调的是，“子类”对“父类”的继承绝不是简单的重复，正如人类的继承一样，总是有发展的。而且，一个“子类”也不仅只有一个“父类”继承。而有时还有多个“父类”的特征，等等。

因此，面向对象认识有三个要点：归类地认识事物，继承地认识事物及发展地认识事物。这三个要点说明，对事物必须是联系起来看，历史地看，发展地看。方法论与认识论历来是统一的。弄清面向对象的认识，将要有助于弄清面向对象分析、设计及编程。

关于面向对象，有人说，早在 20 世纪 50 年代，诸如“对象”和“对象的属性”这样的概念，就出现在人工智能的著作中。OO（Object-oriented，面向对象）的实际发展却是始于 1966 年。当时 Kisten Nygaard 和 Ole-Johan Dahl 开发了 Simula 语言。Simula 提供了比子程序更高一级的抽象和封装；为仿真一个实际问题，引入了数据抽象和类的概念。大约在同一时期，Alan Kay 正在尤他大学的一台个人计算机上努力工作，他希望能在其上实现图形化和模拟仿真。尽管由于软硬件的限制，Kay 的尝试没有成功，但他的这些想法并没有丢失。20 世纪 70 年代初期，他加入了 Palo Alto 研究中心（PARC），再次将这些想法付诸实施。

面向对象与面向过程相比，面向对象更合乎人认识世界的规律的，是较为切合实际的认识世界与改造世界的方法。有人讲，这是小孩也都是这么认识事情的，没有什么神秘的地方。既然如此，计算机编程、PLC 编程完全应该与可以运用它。

2. 面向对象编程（OOP）

OOP（Object-oriented Program，面向对象编程），是用面向对象认识与方法进行的编程，是计算机编程最先进的方法。可把它与结构化编程相比较，在比较中认识。

SP（Structure Program，结构化编程），是从系统的功能入手，将系统分解为若干功能模块，而系统则是这些模块的集合。它强调功能，认为，程序 = 算法 + 数据结构，而数据与代码则是分离的。SP 的优点是，系统是由简单部分组成，可使编程简化。但由于用户需求的变化和软、硬件技术的发展，划分的模块将不稳定。这样，开发出来的模块可重用性不可能高。

OOP 则是以数据为中心，并把数据、代码封装在一起。强调的是数据，即对象及对象间的关系。它认为，程序 = 对象 + 对象之间的关系。数据相对于功能，更稳定些。所以，它的程序可重用性要好得多。由于面向对象编程的可重用性好，可以在应用程序中大量采用成熟的类库，从而提高编程效率，缩短开发时间。此外，它的封装，也可使数据的安全得以保证。它的继承和多态还使应用程序的修改带来的影响更加局部化，更易于维护、更新和升级。

OOP 一般要用到面向对象的编程软件，如 JAVA、VC、DELPHI、VS. NET 等。在编程过程中，可使用编程软件提供的类库（如 MFC）、WINDOWS API 函数、各种控件，以及导向模板，编程可得到很大简化，但要用到类、对象、封装、继承及多态这些概念。而如果使用组态软件，如世纪星、组态王、力控等，也是用面向对象编程，但对用户用不到类、对象、封装、继承及多态等这些概念，编程要简单得多。

3. 面向对象方法

20 世纪 80 年代末以来，OOP 的发展促使 OOD（Object-oriented design，面向对象设计），进而 OOA（Object-Oriented Analysis，面向对象分析）的出现与发展。

OOA 是用面向对象的方法去分析程序所要服务的系统，要从问题域词汇中，通过发现的类和对象，来考察需求的分析方法。OOA 的结果是一系列从问题域导出对象，还要确定基本的对象行为。它的侧重点是业务领域分析，与程序所要应用的行业领域相关，一般由用户方的专家进行。

OOD 是用面向对象的方法进行系统设计。目的是为真实世界建立一个计算机中的虚拟模型。它以 OOA 的成果为基础，结合实际情况出发，而且没有一定的不变的步骤。也不存在界限分明的 OOD 阶段。事实上，OOD 还与 OOP 交织在一起的，没有编程作基础，无法进行 OOD，也无法改进 OOD。

OOD 的精髓就在于锻炼一种计算机式的解决问题的思维方法，这是开发软件的一把万能钥匙，不同的开发工具只带来实现效率的不同。负责 OOD 工作的人被称为系统架构设计师的任务是：

（1）确定系统的总体框架——大多采用已有的领域框架；

（2）正确理解需求分析得出的领域模型，用面向对象的思想设计出软件体系结构——系统概要设计；

（3）分析现实的可获取的技术资源，分解出程序的各个组件，安排好开发任务流程——系统详细设计。

怎么处理 OOA、OOD 及 OOP 间的交互关系，及安排它们具体进程，诞生了几十种的软件开发的面向对象方法。只是这些与 PLC 编程关系不大，在此略。

9.5.2　PLC 面向对象编程设想

以上讲的面向对象都是针对计算机的。PLC 怎么样？能不能设想 PLC 也用面向对象的方法编程？能不能也使用 OOA、OOD？有这个可能。而且有的已在使用。

从历史看，计算机的昨天，也就是 PLC 的今天。在过去的几十年中，计算机编程语言的进化，从地址（机器语言）到名字（汇编语言），到表达式（第一代高级语言，如 Fortran），到控制（第二代高级语言，

如 Cobol)，到过程和函数（第二代和早期第三代高级语言，如 Pascal)，到模块和数据（晚期第三代高级语言，如 Modula)，最后到对象（基于对象和面向对象的语言)。难道计算机今天已做到的事情，PLC 明天就做不到吗?

再看看 DCS。它的硬件组成比 PLC 要复杂得多。但软件编程，却不比 PLC 复杂。为什么？因为 DCS 厂商在提供硬件的同时，也提供相应的软件包。在此基础上编程，多数只是选定对象或组件，再填写参数，连线也就完成了编程。DCS 能做到的，PLC 为什么不能?

组态软件也一样。从本书第 7 章的介绍知，用它编程，比用其它语言编程要简单得多。原因是它用了高水平的面向对象方法。只要选用好它提供的对象，就完成了编程。面向对象最大的优点是程序能够重用。程序可重用，就可把编程当成产业，建立专门“生产软件的工厂”生产软件。用户像买商品一样买去、重用。当今的组态软件不正是这样吗?

最后谈谈嵌入式系统。沈阳鹭岛公司开发的 LEODO 人机界面，它的编程，也是面向对象的。只是它先在计算机上开发，编译后再下载给 LEODO。

所以，PLC 也应是这样。PLC 厂商在提供硬件的同时，也应尽可能的提供相应软件。这软件也应是面向对象的，可重用的。事实上，PLC 的这种局面也已出现。以 OMRON 公司为例，在推出新的硬件模块的同时，有的也推出可进行面向对象编程的软件。用户用其作再开发，即可成为自己很好的应用。如在它新推出的回路控制模块的同时，也推出这个回路控制模块的编程软件，这在本书第 4 章的介绍中，读者已经看到用它编写模拟量控制程序多么简单。

其实，现也有 PLC 厂商的编程工具软件，也已引入对象概念。也有特性、方法之类东西。不过不太普遍而已。

这里也有个观念问题。人们一般舍得花钱买硬件，但舍不得花钱买软件，这观念不更新，软件业就无法发展。特别对 PLC 这样，先前多是只提供硬件，而不提供软件的行业，更是如此。

然而，如果 PLC 硬件厂商如提供硬件一样，也能提供可重用的 PLC 程序（软件)。用户重用厂商提供的程序（软件)，只要做些设定，做些软件模块间的连接，填填参数，就可成为自己的程序，那该有多好。

计算机编程已经出现了这个趋势，这也是面向对象编程发展的结果。计算机软件将组件化。所需要的组件可定购。软件，像定购硬件一样也可定购。软件可安排在“软件工厂”用“白领工人”“生产”。用户只要配备软件系统专家，主要是做 OOA，OOD 工作。按 OOD 的要求去订购软件。最后按 OOP 的原则，组装所购软件，再组成为自己的程序。美国大的工业公司原来编程人员很多，因为，所有软件都要自己开发。而今，有了面向对象的组件化软件，或可订购通用的软件，就不必什么软件都自己编了。现在这些公司不少在裁减编程人员，与这个软件的发展趋势是不无关系的。

自然，要实现这个目标，要做的工作很多，要走的路也很远。但对 PLC 而言，可以这样设想:

(1) 引入面向对象的有关概念。在 PLC 程序的需求分析、算法设计及编写代码编程中，引入面向对象的有关概念，也使用类、对象、封装、继承、多态等有关方法进行处理。其实，现在有的 PLC 编程软件已引入对象这样概念了。

(2) 根据对象的含义定义数据。是否可把 PLC 的输入、输出点，如 0.00，10.00 这样数据也都看成是对象？这些点的集合就是“输入、输出点”的类。这个类有其特性，如处于 ON 或 OFF 状态。还有处理它的方法，如使其变为 ON 或 OFF（在一定的条件下，用 OUT 或 OUT NOT 的指令作用于它)。

这些输入点还可分为普通输入点、高速计数输入点。高速计数输入点还可分为若干不同类型，以用于不同的模式的高速计数。

如果像计算机编程，做好变量定义后，不要再作很多繁琐的设定，PLC 就能为之自动进行软设定，并能即时设定即时起作用。

有的方法可否也能按多态性方法处理？如把高速计数方法用于某可进行高速计数的输入点，则自动将其作高速处理。如把普通的输入用于某可进行高速计数的输入点，则自动按普通输入点处理，等等。

(3) 硬件软件一体化。PLC 高功能模块越来越多，功能也越来越强。但最好在推出这些硬件模块的同

时，也推出使用它的软件。它的设定、使用程序，能在可视化的编程界面上用面向对象的方法完成。而PLC 只是在建立工程时，把这些特殊模块的使用程序组织进来就是了。

（4）改进 PLC 编程语言。现在虽说 PLC 有 5 种语言。但实际上像本书一样，多数只用梯形图语言，或助记符语言。用梯形图语言、助记符语言进行面向对象编程比较难。较便于面向对象编程的图形符号语言是功能块图语言，文字符号语言是结构化文本语言。这两种语言应大力推广，使其普及。听说 OMRON CX-Programmer 已增加有这种功能块图语言。

DCS 用的多为功能块图。只要选选功能块，填填参数，接接连线，就可完成一组程序的设计。而这些功能块由 DCS 厂商提供，用户也可开发。可不受限制的重用，是很方便的。

计算机面向对象编程用多是文本语言，如 VC、DELPHI 等等都是文本语言。但有 MFC 库、API 函数、种种控件，再加上建立过程的种种导向模板，所以，虽说是文本语言，但要写代码量也少了很多。显然，PLC 要推广面向对象的文本语言，也必须这么作。

当然，如同自然语言的演变只能是渐进的、漫长的一样，PLC 编程语言进步也不能一蹴而成。只能一步步向面向对象编程语言推进。

（5）优化 PLC 编程软件。编程语言变化了，概念变了，当然编程的工具软件也得变。现有的 PLC 厂商的编程软件虽都不断地在进步，每年都有新版本推出，但总的看能与微软的 WORD、VB、VC 界面相当的 PLC 编程软件是不多的。多数只能与“写字板”相当。有不少也只是“记事本”水平。这样的编程软件状态怎么能支持面向对象编程?

（6）建立开放的机制。编程软件应是开放的。允许其它方或用户也可开发像 DLL、ACTIVEX 那样，被嵌入 PLC 程序中使用。

以上设想是在本书第 1 版中提出的。只过了 2 年，令作者惊喜的是，国内生产 PLC 的和利时公司 LM 小型 PLC 编程，不少地方与上述设想“巧合”，也使我们看到了 PLC 面向对象编程的亮点。特别要提到的它有如下这么 4 点：

（1）LM 型 PLC 机编程已引入了面向对象的有关概念，并可以按照对象的理念声明变量。它没有常规 PLC 那么多的内部器件，如定时器、计数器，而代之以变量。这些变量按需要声明，使用多少，就声明多少。变量名还可以按照其功用命名，比起器件编号更便于辨认。变量还可以分为全局的与局部的、掉电保持的与不保持的、与内部器件关联的与不关联的，都可以用变量声明选定。同时，变量的类型还很多，计算机有的，包括结构、枚举、数组、指针，它都有。这些不仅可以为编程提供方便。也使 PLC 内存得到充分的利用。同时，还可以做到程序代码与程序数据紧密结合，确保程序的安全。

至于它有的输入、输出点已经可以分为普通输入点、高速计数输入点，并可以调用功能块实现普通输入、高速计数输入或普通输出、脉冲链输出或可调脉宽输出。这样，就不要再做很多烦琐的设定，就能“即调用即起作用”。

（2）LM 型 PLC 编程实现了硬件、软件一体化。它没有庞大的系统设定区，也没有复杂的 PLC 设置窗口。它的设定主要用软件功能块。有一种硬件功能，就有与其对应的软件功能块。调用这些功能块就相当于传统 PLC 的设定。如串行接口通信参数的设定，可以调用串行接口参数设定功能块。如果要回到系统默认设定，可以调用恢复系统默认参数的功能块，非常方便。

（3）LM 型 PLC 编程软件完全合乎国际的自动化编程软件的标准。它可以使用 6 种编程语言。不仅可以用梯形图语言，或助记符语言，还可以使用较便于面向对象编程的功能块语言及结构化文本语言，这在现有的 PLC 中是不多见的。

（4）LM 型 PLC 编程已经建立开放的机制。其指令系统由基本指令、函数及功能块组合而成。后两者是由库文件提供，可根据需要加载到程序中，用多少就加载多少。PLC 的程序内存也因此得到更有效地利用。种种功能丰富的库文件由和利时公司来提供，而用户自已也可以生成。因而在某种意义上讲，它的指令系统是无限大的。只要 PLC 的内存允许，要什么指令就可以生成什么指令，要多少指令就有多少指令。

这几年，OMRON 公司在这些方面的进步也是明显的。它现在也可使用功能块编程。OMRON 公司提供

有系统功能块，用户也可设计自身的功能块。在它推出新硬件的同时，也都推有软件。这些也都为推进面向对象编程提供新的支持。

9.6 PLC 程序调试

关键词：Debug、程序调试、程序仿真测试、程序联机测试、程序现场测试、程序定型、程序保护、程序加密、程序加锁、程序评价

9.6.1 PLC 程序调试概述

1. 程序调试目的

PLC 程序调试，英文称 Debug，找“虫子”，其含义就是，测试程序，查找错误，改正错误。直到程序合乎要求，能圆满地实现程序的功能为止。

一个初编的程序没有错误是绝对不可能的。有人讲，成功的程序不是编出来的，而是调出来的，这不无道理。

2. 程序调试内容

PLC 程序调试的内容有：语法检查，语义检查，输入、输出检查，逻辑效果测试，各个功能测试。

语法检查在脱机状态，即不用与 PLC 联机即可进行。在用编程软件送入指令时，一般就会进行语法检查。一旦语法出错，将有提示，以至于难以继续编下去。语法错误多是对指令理解不正确，或指令的操作数使用不当造成的。在进行程序编译时，编程软件还会进行全面的语法检查。如有错，将有很详细的提示。

语义检查比较难。这类错误的特点是，语法正确，但不能产生所期望的结果。对它的测试，可按指令类型分组进行。以确保所使用的指令及其操作数能达到预期的目的。

输入、输出检查，则是更大范围的调试。主要是观测输入与输出间的对应关系是否与预期的相同。这种测试，不仅用基本的输入数据，还要用多组数据。以至于非正常的情况也可作些测试。

逻辑效果测试及各个功能测试，这是更全面的测试。通过这些测试，可以查清程序能否达到用户的使用要求。

3. 程序调试类型

PLC 程序调试的方法有：脱机调试，仿真调试，联机调试及现场调试。

（1）如果用编程器编程，脱机调试是指，编程器与 PLC 联机（有的 PLC 可不联机），使 PLC 处编程状态，向 PLC 键入一条条指令操作码及操作数，并进行语法检查。直到纠正所有的语法错误为止。

如果用计算机编程，脱机调试是指，在计算机上，运行 PLC 厂商提供的编程软件，不与实际的 PLC 或模拟的 PLC 联机所进行的调试。也是先键入一条条指令操作码及操作数，进行语法检查，直到纠正所有的语法错误为止。

（2）仿真调试是在计算机上，用模拟软件进行。不仅可检测语法错误，还可检测语义错误。

（3）联机调试，不一定在工业现场，但必须有 PLC，并与联机，进行调试。它是真刀

真枪的调试，可发现及纠正的程序更多些。

（4）现场调试是指，到工业现场进行的程序最后调试。而且，只有这一步调试通过，编程的工作才能算是成功。

9.6.2　PLC 程序仿真调试

PLC 程序仿真调试指用厂商提供的仿真软件代替实际 PLC，在计算机上所进行 PLC 程序调试。

仿真调试的关键得有仿真软件。OMRON 公司 CS1、CJ1 型 PLC、CP1H 及后续机型，该公司都针对其配备有仿真软件，CX-Simulator V1.7 版本。使用它，即使无实际 PLC，也可进行程序调试。CX-Simulator V1.7 集成在 CX-one 软件包中。CX-one 安装后，它与编程软件 CX-Programmer 也一并安装。并在原 CX-Programmer 软件的 PLC 编程菜单中的“在线模拟”项将激活，如图 9-18 所示。



图 9-18　CX-Programmer　菜单

1—PLC 菜单项下“在线模拟”项激活　2—工具条上“在线模拟”项激活

编好 PLC 程序后，用鼠标单击“PLC 编程”项，再选择击“在线模拟”项，也会弹出程序下载窗口，同时还出现“CX-Simulator debug cosole”窗口，后者如图 9-19 所示。

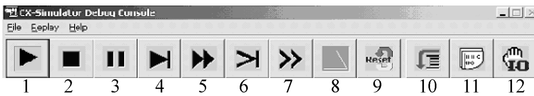


图 9-19　仿真 PLC 窗口

1—运行　2—停止　3—暂停　4—单步执行　5—连续步执行　6—单扫描执行　7—多扫描执行　8—扫描代替　9—复位　10—显示步运行　11—任务管理　12—I/O 管理

从图 9-19 知，该窗口仅是一些操作键。程序、设定等下载后，可单击图中的“运行”键，仿真 PLC 即进入运行状态。在 CX-Programmer 软件上看到的以及对其的操作和与真 PLC 联机时完全相同。

更方便的是，使用仿真器调程序，还可单步调。办法是单击图“显示步运行”键。击后，将弹出“Step Run”窗口，如图 9-20 所示。

从图 9-20 知，该窗口显示的是一条条助记符指令。点击图 9-19 的“单步执行”键，程序将一步步执行。击一次，执行一条指令。如图 9-20 所示，它执行的是地址 4 的指令。如

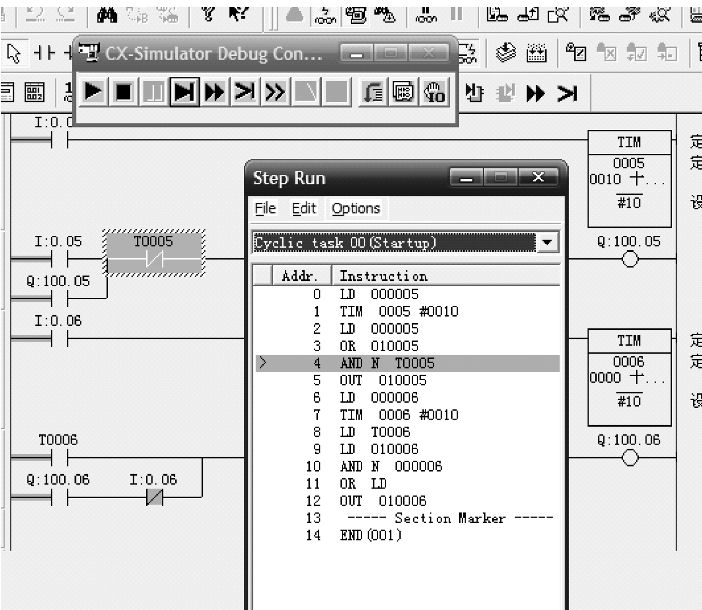


图 9-20 “Step Run”窗口

图所示，不显示“Step Run”窗口也可在梯形图上显示步运行。点击“连续步执行“键。可连续一步步执行。点击“单扫描执行”键，可一步步执行一个扫描周期的所有指令。

这个“单步执行”、“单扫描执行”或“连续单步执行”，对理解程序，观察程序的执行过程是很有益的。

OMRON 公司的这个仿真器，“仿”PLC 仿的相当真，相当难得。对编程和学习 PLC 都是非常好的工具。

此外，Simulator 也可自身进入。软件安装后，会在开始菜单中，显示有“OMRON”\ CX-one \“CX-SIMULATOR”\“CX-SIMULATOR”。

点击“CX-SIMULATOR”，将弹出有关设定、网络链接及仿真 CPU 单元。图 9-21 所示即为这个 CPU 单元。用它，再加上 CXP 软件，将仿真得更为逼真。

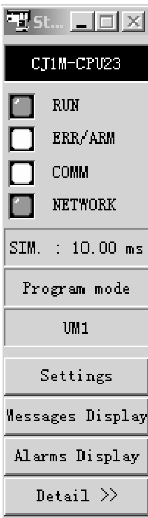


图 9-21 仿真 CPU 单元

此外，还可进行一些网络仿真。在此，就不多介绍了。

9.6.3 PLC 程序联机调试

PLC 程序联机调试要用实际 PLC，但未与实际设备相连。目的也是检查程序的正确性。联机调试最关键的是计算机能与 PLC 通信，能下载程序，设定，并对 PLC 进行操作，还能查看 PLC 的状态和数据。

联机调试是程序调试不可逾越的阶段。仿真虽好，但只是“纸上谈兵”，联机才是真正应用，更何况有的 PLC 还不能仿真。

有关联机调试的一些细节，在第 2 章介绍 CX-Programmer 软件时已有说明，有关内容不再重复。只是强调两点：

（1）如程序较大，最好按功能分块，分块调，分块去实现所预想的功

能。分块的办法是利用 END 指令及“段”位置的上下调整。记住，OMRON PLC END 之后的指令是不被执行的。

（2）调试程序要稳步推进。调试一段成功了，要注意存储，把成果巩固下来。修改程序，修改前的也要先存。以避免程序改乱后，退不到原来已取得的成果。

有关程序的逻辑关系，在联机调试完全可以弄清，应着力在联机调试时予以解决。至于系统的动力学问题，如 PID 程序，参数选择是否适当，由于系统未接入工作，因此就看不出来了。那只好在现场调试中解决。

9.6.4　PLC 程序现场调试

PLC 程序现场调试是在工业现场进行。所有设备都要安装好后，所有连线都要接好，是 PLC 程序的最后调试。

现场调试通过后，可交给用户使用，或试运行。

现场调试参与的人员较多，要组织好，要有调试大纲。要依大纲，按部就班地一步步推进。开始调试时，设备可先不运转，甚至也不要带电。可随着调试的进展，逐步加电、开机、加载，直到按额定条件运转。具体过程大体是：

（1）要查接线、核对地址。要逐点进行，要确保正确无误。可不带电核对，那就是查线，较麻烦。也可带电查，加上信号后，或产生输出后，看 PLC 上的指示灯的变化。

（2）检查模拟量输入、输出。检查输入模块、输出模块设定是否正确，工作是否正常。必要时，还可用标准仪器检查输入、输出的精度。

（3）检查与测试指示灯。控制面板上如有指示灯，应先对指示灯的显示进行检查。一方面，查看灯坏了没有；另一方面，查逻辑关系是否正确。指示灯是反映系统工作的一面镜子，先调好它，将对进一步调试提供方便。

（4）检查手动动作及手动控制调试逻辑关系。要查看各个输出点，是否有相应的输出以及与输出对应的动作，然后再看各个手动控制是否能够实现。如有问题，要予以解决。

（5）半自动工作。如系统可自动工作，那也要先调半自动工作能否实现。调试时可一步步推进。直到完成整个控制周期。哪个步骤或环节出现问题，就着手解决那个步骤或环节的问题。

（6）自动工作。在完成半自动调试后，可进一步调试自动工作。要多观察几个工作循环，以确保系统能正确无误地连续工作。

（7）模拟量调试、参数确定。以上调试的都是逻辑控制的项目。这是系统调试时首先要调通的。这些调试基本完成后，可着手调试模拟量、脉冲量控制。最主要的是选定合适控制参数。一般讲，这个过程是比较长的。要耐心调，参数也要作多种选择，再从选出最优者。有的 PLC 的 PID 参数可通过自整定获得。但这个自整定过程是需要相当长的时间才能完成的。

（8）异常条件检查。完成上述所有调试，整个调试基本也就完成了。但最好再进行一些异常条件检查。看看是否会出现异常情况，或一些难以避免的非法操作，如可能出现问题，也应予以解决。

提示：在现场调试中程序的变动，包括功能的增减、变化是难免的。程序是否好用最终也只能通过现场调试才能确定。所以，要重视现场调试，并力争通过现场调试，使程序

进一步完善。

9.6.5 PLC 程序文档

通过现场调试的程序，需要在计算机中定型，并存储在磁盘或光盘中。此外，还要建立必要的程序文档。

这个程序文档，除了源程序代码，一般还应包括插入到程序代码中的注释和专门制作的文件。这是 PLC 程序编写、调试成功后必须完成的一个重要工作。程序交付给使用方，也应当交付这个文档。

注释是插入到 PLC 程序代码行中的解释性注解。有了注释的程序比较容易理解。在什么地方该加注释，并没有固定的法则。大体上讲，一个文档比较好的程序，都包含解释程序目的注释，各程序段的功能注释，特殊指令的使用注释等等。在一些意思不太明确的地方段也应加上注释。

专门制作的文档不属于程序，是一些编程和程序使用的有关信息。文档既可以是文字的，也可以是电子格式的。一般有 2 个：程序手册和用户手册。

程序手册包含所有编程的有用信息，包括问题描述和算法。程序手册是软件开发和维护的工具。设计者才用，用户不用。

用户手册包含用户使用程序的所有信息，可以帮助用户学会使用该 PLC 程序。用户手册的电子版本通常还能给用户提供在线式帮助。

编程人员在编制程序的同时，一般要负责编写程序手册及用户手册。后者也可由专业的科技作家去编写。科技作家专门解释科技概念和程序，把复杂的概念简化，非技术人员阅读时，将更便于理解。

9.6.6 PLC 程序保护

程序保护可使用读写 DIP 开关，ON 保护，否则不保护。也可用软件设定保护，如 CPM 型 PLC 是 DM6602 字的 0 位，设为 1，保护，设为 0，不保护。程序保护可保证程序不被删除或修改。但其他人可读它，重用它。为了保护知识产权，可对程序加密。

OMRON C 系列 PLC 加密的方法是：

在程序的 00000、00001 地址分别加入如下指令：

```
0000   LD   AR1001
0001   FUN（49）
       000
       000
       #密码
```

这里，第 1 条指令为加载指令（LD），其操作数为 AR1001。第 2 条指令为功能指令（FUN（49）），其操作数有 3 个，即 000，000 及#密码。

密码可从 0000 ~ FFFF（十六进制）数码中任意选定。

以上是整个程序加密。也可在程序某地址之后加密。办法是在加密开始处输入如下指令：

开始处地址　LD　AR1002

开始处地址 +1　FUN（49）
000
000
#密码

已加密的程序，用计算机编程软件打开时，要先输入正确的密码，否则读不出程序。使用简易编程器，也要先解密，否则也读不出这个程序。简易编程器进行解密操作，见第 2.5 节。

OMRON CS 系列 PLC 加密的方法是用编程软件。在程序属性窗口中设定。密码为 8 位，可以是数字，也可大小写英文字母。

程序除了保护、加密，还可加锁。加锁后，即使 PLC 程序正常运行，但不产生控制输出。加锁可用置位 PLC 的输出禁止位实现，也可用自编一段小程序，使相应的输出禁止。

提示：程序保护、程序加密、程序加锁是不同概念，各有各的目的。可按要求全用或用其中的一种。

9.6.7　PLC 程序解密

程序解密要使用原来密码。如果忘记密码，4 位加密与 8 位加密处理方法不同。

4 位密码加密的可清除程序内存，并下载新编程序。也可用解密软件解密。

8 位密码加密的无法简单清除程序，也无法下载新程序。解决办法是，在另一台 PLC 上，编写一个无密码保护的程序，然后使用一块 CF 卡，用 Autoexec.obj 和 Autoexec.std 文件名，将程序和参数存储在 CF 卡上。然后，把此卡取出，安放在要解密的 PLC 上，并将其 CPU 的 DIP 开关 2 置为 ON，再给要解密的 PLC 上电。这时，PLC 的加密的程序将被 CF 卡上的程序替换。这时，密码将解除，但原来加密的程序将永远看不到了。

9.6.8　PLC 程序评价

要判定 PLC 程序的质量如何，最好的评价标准是实践，看程序能否达到预期的目的。但这还不够。因为能达到目的程序还有好与不好之分。到底什么样的程序算是好程序？定性地讲，大体有这么几个方面。

1. 正确

PLC 的程序一定要正确，并要经实际工作验证，证明其能够正确工作。这是对 PLC 程序的最根本的要求，若这一点做不到，其它的再好也没有用。

要使程序正确，一定要准确地使用指令，正确地使用内部器件。程序出现差错，或不正确，往往与这两点有关。

准确使用指令是与准确理解指令相联系的。为此，对指令含义、使用条件一定要弄清。必要时，可编些小程序对一些不清楚的指令作些测试，以求弄明白。

值得注意的是，同一指令，由于 PLC 的出厂批次不同，在个别情况下，一些细节有可能不完全一样，应弄清。

再就是，有的指令执行一次就可以了，多了反而不行。有的则要一直执行。有的指令在子程序中执行与在主程序中执行情况可能不一样，等等。这些细节也要弄清，不弄清也很容易出错。

内部器件正确使用也是重要的。如有的有掉电保护，有的没有。一定要做到该掉电保护的一定要用掉电保护的器件，反之则不能用。定时器、计数器前边的号（随机型而异），有的 PLC 小于 15 号的可中断计时，可用于精确计时。若用精确计时，用的不是能中断工作的定时器，其计时精度将无法保证。等等。

总之，要准确使用指令，正确使用内部器件，使所编的程序能正确工作，这是对 PLC 程序最根本的要求。

2. 可靠

程序不仅要正确，而且要可靠。

可靠反映着 PLC 程序的稳定性，这也是对 PLC 程序的基本要求。

有的 PLC 程序，在正常的工作条件下，或合法操作时能正确工作，而出现非正常工作条件（如临时停电，又很快再通电），或进行非法操作（如一些按钮不按顺序按，或同时按若干按钮）后，程序就不能正常工作了。这种程序，就不大可靠，或说稳定性不好，就不是好的 PLC 程序。

好的 PLC 程序对非正常工作条件出现，能予以识别，并能使其与正常条件衔接，可使程序适应于多种情况。

好的 PLC 程序对非法操作能予以拒绝，且不留下“痕迹”。只接受合法操作。

实现程序可靠的方法靠记忆与连锁。PLC 一些内部器件，可掉电保持，有记忆功能。所以，靠它可使程序在临时断电再复电后执行，仍可衔接。

联锁是拒绝非法操作常用的手段，继电电路常用这个方法，PLC 也可继承这个方法。如控制某电动机正反转的两个按钮，同时按下，显然是非法操作。拒绝接受这个非法操作的办法是，把这两个按钮的非信号加入到对方的工作条件中，以实现联锁。等等。本书第 8 章讨论的 PLC 程序的可靠性，讲的都是这些问题。

总之，使 PLC 程序能可靠工作，是很重要的。

3. 简短

使 PLC 的程序尽可能简短，也是应追求的目标。

程序简短，可节省用户存储区；多数情况下也可节省执行指令时间，提高对输入的响应速度；还可提高程序的可读性。

程序是否简短，一般可用程序所用的指令条数衡量。用的条数少，程序自然就简短。

要想程序简短，从大的方面讲，要优化程序结构，用流程控制指令简化程序，从小的方面讲还要用功能强的指令取代功能单一的指令，以及注意指令的安排顺序，等等。这些在以后还要作相应介绍。

4. 省时

程序简短可以节省程序运行时间，但简短与省时并不完全是一回事。因为运行程序时间虽与程序所拥有指令条数有关，而且还与所使用的是什么指令有关。PLC 指令不同，执行的时间也不同。而且，有的指令，在逻辑条件 ON 时执行与在 OFF 时执行其时间也不同。另外，由于使用了流程控制指令，在程序中，不是所有指令都要执行。所以，运行程序的时间计算是较复杂的。但要求其平均时间少、最大时间也不太长是必要的。这样，可提高 PLC 的响应速度。

省时的关键是用好流程控制指令。按情况确定一些必须执行的指令，作必备部分，其余

的可依程序进行，有选择地执行，或作些分时工作的设计，避免最大时间太长，等等。

程序执行时间可作计算，也可实时测试。如 C200H 型 PLC，其中辅助继电器 AR26 通道，就存储着当前周期的运行时间，AR27 存储着周期最长的时间。查这个存储单元的内容，即可知道所需的运行时间。

5. 可读

还要求所设计的程序可读性要好。这不仅便于程序设计者加深对程序的理解，便于调试，而且还便于别人读懂程序，便于使用者维护。必要时也可使程序得以推广。

要使程序可读性好，所设计的程序就要尽可能清晰。要注意层次，实现模块化，以至于用面向对象的方法进行设计；要多用一些标准的设计。

再就是 I/O 分配要有规律性，便于记忆与理解。必要时，还要做一些注释工作。内部器件的使用也要讲规律性，不要随便地拿来就用。

可读性在程序设计开始时就要注意。这不易完全做到。因为在程序调试的过程中，指令的增减，内部器件使用的变化，可能使原较清晰的程序变得有些乱。所以经调试的程序，最好再作些整理，在可读性上做些文章，以保证所设计的程序具有更高的质量。

6. 易改

还要使程序便于修改。

PLC 的特点之一是方便，可灵活地适用于各种情况。其办法就是靠修改或重新设计程序。

重新设计程序用于改变 PLC 用途的情况，不仅程序重编，而且 I/O 也重新分配。多数情况不须重编程序，作一些修改就可以了。这就要求程序具有易改性。

易改也就是弹性，要求只要作很少的改动，即可达到改变参数或更改动作的目的。

为此，要使程序尽可能循序推进，采用步进控制的方法。一个动作到另一个动作转换靠转换控制步实现。更改时，只要更改步的内容，而不必更改整个逻辑。参数的设定尽可能用间接的方法。如时间继电器的定时值，其参数不直接设定，用指定某内存单元的内容设定。而这个内存单元的单元或靠程序初始化时（起动时第一个扫描周期）赋值，或由编程器（或操作器）临时设定。这样，要作更改，只须改有关内存单元的内容也就可以了，比较灵活。怎么使所设计的程序富有弹性，便于修改，办法多得很。但最主要的是，在选用指令时，注意留有余地，要考虑到，如果有了变化，能不能改，以及怎么改。只要注意了这个问题，设计出较有弹性的程序是完全可能的。

这 6 条是作者在 1998 年出版的,《可编程控制器配制　编程　联网》一书中提出的没有量化指标。其实，对计算机编程，软件的质量有量化指标。转录如下，不妨可作为评价 PLC 程序的参考。

常见到的软件质量指标有 10 项：

（1）平均不工作间隔时间（Mean Time Between System Downs，MTBD）。设 TV 为软件正常工作总时间，d 为系统由于软件错误而停止工作的次数，则有 $MTBD = TV/(d+1)$。

（2）系统不工作次数（在一定时间内）。由于软件错误停止工作，必须由操作者去介入再启动才能继续工作的次数。

（3）平均故障修复时间（MTTR）。它反映了出现软件缺陷后采取对策的效率。

（4）平均故障停机时间（Mean Down Time，MDT）。由于软件错误，系统停止工作时间

的均值。

（5）可用性 A。设 TV 为软件正常工作时间，TD 为软件错误使系统不工作的时间，则定义

$$A = TV/(TV + TD)$$

上式也可以表示为

$$A = MTBD/(MTBD + MDT)$$

（6）初期错误率。一般以软件交付使用方的三个月为初期错误率期。初期错误率以每 100h 的错误数为单位。它用来评价软件交付使用时的质量，并且预测何时软件可靠性基本稳定。

（7）偶然错误率。一般以软件交付给使用方后的四个月，为偶然错误期，偶然错误率一般以每 1000h 的错误数为单位，它反映了软件处于稳定状态下的质量。一般要求偶然错误率不超过 1，即每 1000h 不到一个错误，亦即 MTBF 超过 1000h。

（8）使用方误用率。使用方不按照软件说明书等文件进行使用所造成的错误叫使用方误用。在总使用次数中，使用方误用次数所占的百分率叫做使用方误用率。

（9）用户提出补充要求数。如果用户对软件提出了补充要求，则反映了软件的功能尚不能充分满足用户的需要。如果这种情况在软件应用了一段时间以后频繁出现，则说明软件已经进入老化期，应该开发新的软件来取代它。一般在偶然错误期，用户每月提出的要求数不应超过 1。如果平均数超过 1，则认为软件已进入老化期。

（10）处理能力。处理能力有各种指标。例如可以用每秒钟处理多少输入变量、更换每一幅 CRT 画面需要几秒钟等来表示。

只是，对 PLC 而言，程序的用户总是一个或有限的几个。不像计算机软件，要面对很多人和很多情况。评价指标便于比较和统计。而且，PLC 任何软件出错的后果都可能是严重的，也都是不允许的。所以，以上 10 条仅供参考。

结语

本章讨论的程序组织，目的是使若干小程序模块、设定模块及其它数据模块进行整合，进而组成一个工程，以便于管理。这也是 PLC 编程技术发展的新趋势。过去，PLC 程序比较简单，一个程序模块就够用了。而今，趋向于多段（Section）、多任务（程序）、多 CPU 的组织程序更便于分工合作编程，加快编程的进程。

另外，为程序调试也提供有很多有效的方法。如仿真、单步调试、制定扫描次数调试、强制置位、复位等。应充分利用好这些方法做好程序调试。

学习本章，又可学习一些新软件，对 PLC 资源又将有新的认识。

请想想

1. 程序组织含义？
2. 程序模块化组织含义？
3. 程序多任务（多程序）组织含义，优点？
4. 多 CPU 系统程序组织特点？
5. 程序柔性化含义及要点？

6. 面向对象编程含义及其前景？
7. 程序调试要点？

请试试

1. 设计一个模块化程序。
2. 设计一个多任务（多程序）组织程序。
3. 设计一个柔性化程序。
4. 其它实际控制程序设计。

第 10 章　PLC 在前进

PLC 诞生之后，经历了 20 世纪 70 年代中期发展，到 70 年代末期进入成熟阶段。不仅技术向大规模、高速度、高性能发展，而且产品也走向系列化，高、中、低档产品日渐齐全。应用范围也日趋扩大，使 PLC 成为一个新兴的产业，产值与销售量都在不断地增加。1982 年，日本有 40 多个厂家生产 PLC，品种有 120 多种。1984 年，美国注册生产 PLC 的厂家有 48 个，品种有 150 多种，销售额达 6 亿美元。同年，欧洲有 60 多个厂家生产 PLC，品种近 200 多种。

20 世纪 80 年代以来，PLC 发展更为迅速。不到 30 年时间，各主要厂家都已更新或换代过多种系列的产品，大体每 3 ~5 年就有换代的，年年都有更新。销售额年增长率一直保持为 30% ~40% 。1993 年，销售额达 39 亿美元，2000 年销售额达 76 亿美元。几乎都已占了该年整个自动化控制系统产品（含 PLC、DCS、IPC 等）市场份额的一半。已名副其实地成了控制技术的重要支柱。

目前全世界有 200 多厂家生产 300 多品种 PLC 产品，主要应用在汽车（23% ）、粮食加工（16.4% ）、化学/制药（14.6% ）、金属/矿山（11.5% ）、纸浆/造纸（11.3% ）以及其它（23.2% ）行业。

我国的 PLC 研制、生产、使用也发展很快。特别在使用方面，在引进一些成套设备的同时，也引进不少 PLC。如上海宝钢第一期工程，就采用了 250 台，第二期也采用了 108 台。再就是秦皇岛煤码头第二期和第三期、天津化纤厂、秦川电站、北京吉普汽车生产线、西安的彩电和冰箱生产线等，也都采用了 PLC。

不少公司，或为国外的公司推销质量与档次较高的 PLC 产品，并负责售后服务；或与国外公司合资，生产各种档次的 PLC，既返销国外，也向国内销售。

目前，国内 PLC 生产企业约 30 多家，年产量超过 1000 台的不到 10 家，在国内的市场份额不到 10% 。

在 PLC 应用方面，我国是很活跃的，近年来每年约新投入 10 万台套 PLC 产品，年销售额 30 亿人民币，应用的行业也很广。

进入 21 世纪后，由于 IT 技术和各种控制技术的迅速发展，PLC 既面临着很多新挑战，而又有很多的新发展。它的性能在不断提高，应用在不断扩展，概念在不断更新，类型在不断增加，正以比过去更高发展速度在前进着。

10.1　PLC 的性能在提高

关键词：工作速度、控制规模、组成模块、内存容量、指令系统、工作可靠、联网能力、外部设备、支持软件、经济效益

PLC 性能指标有两种：一般指标与特定指标。

一般指标规定一些 PLC 的使用条件。同一产家、同一类型的 PLC 则多是相同的。这些

指标有：PLC 的保存与使用温度、湿度，大气气氛，耐电压及绝缘指标，抗干扰指标，抗机械振动、冲击指标，等等。

特定指标规定一些具体型号 PLC，以至于具体模块的具体性能。如某型 CPU 模块，则要指出它的工作方式，指令条数、类型，执行一条指令的时间，可处理的输入、输出点数，内部器件的类型、数量，工作电源类型、电压允许的波动范围，等等。再如某 I/O 模块，则要指出它的点数，信号特性等等。

上述性能指标是比较多的，各产家各有不同的表述，所以，要一一地去介绍它比较繁琐，也没有必要。但从共性的角度，把 PLC 的有关指标归纳为以下 10 个方面是可行的。也可用它，对不同产家 PLC 的发展水平作比较。

10.1.1　工作速度在提升

工作速度指 PLC 的 CPU 执行指令的速度、对急需处理的输入信号的响应速度及通信接口的数据传送速度，但关键是指 CPU 执行指令的速度。

工作速度是 PLC 生命力之所在。如果速度不高，对输入信号响应太慢，PLC 将达不到应有控制效果，更谈不上其它性能的提高了。所以，厂家在发展 PLC 时，首先着眼的是提高速度。

随着执行指令时间的缩短，在允许的扫描周期（一般不大于 100ms）内，可增加运行指令条数，提升处理数据的能力，进而增加 PLC 的控制规模。

随着执行指令时间的缩短，PLC 的中断类型还将增加，其功能也将更加强大。

随着执行指令时间的缩短，还可实现输入、输出的即时刷新。实质上它也是一种中断。这技术与外中断配合，可使 PLC 对紧急信号的响应时间，可小到毫秒级。大大提高了 PLC 控制的精确度及可靠性。

随着执行指令时间的缩短，也为 PLC 联网通信能力的增强及通信速度的越高提供了可能。

PLC 指令不同，执行的时间也不同。但常以执行一条基本指令的时间来衡量这个速度。这个时间短，说明具体的 PLC 性能好。有的还用每秒能执行多少条基本与传送指令，即 PMX 值，作为它的衡量指标。

以 OMRON 若干机型为例，可看出这个时间的变化。如上世纪 80 年代初的 C60P 型 PLC，执行 LD 指令的时间为 4μs。而到了上世纪 80 年代后期的 C60H 型 PLC，为 0.75μs。上世纪 90 年代的 CQM1 型 PLC，0.4μs；CPM2A 型 PLC，0.64μs；CQM1H 型 PLC，0.375μs。当时的大型 PLC，CV2000，执行 LD 指令仅仅 0.1μs。在当时算是工作速度最快的 PLC。上世纪 90 年代后期，推出的 CS1 型 PLC，此时间已减少到 0.02μs。在当时，其工作速度也算是最快的。进入 21 世纪，新推出的小型 PLC，CJ1H 型 PLC，此时间也只有 0.02μs。

总之，这 20 多年，PLC 的工作速度，大体提高了 200 倍，加快了 2 个数量级。已从“微秒级”，推进到“纳秒级”。

虽然 PLC 的工作速度已有很大提升，但继续提高 PLC 工作速度，还是有潜力的。因为用的同样为微处理器（CPU），目前个人计算机用的最新的 CPU，其工作周期已达到纳秒级，频率高到每秒几个 G。而 PLC 的 CPU 的工作周期还是零点微米级的，工作频率还是低的。所以，还是有很大的提升空间。

总之，工作速度是 PLC 的最重要的，也是第 1 个性能指标。工作速度不断提升是 PLC 在前进的重要体现。

10.1.2 控制规模在扩大

控制规模代表 PLC 的控制能力，一般是看其能对多少输入、输出点（指开关量）及对多少路模拟量进行控制。有时还要看其能扩展多少模块、多少机架、多少站点等等。

控制规模与速度有关。因为规模大了，用户程序也长，执行指令的速度不快，势必延长 PLC 循环的时间，也必然会延长 PLC 对输入信号的响应。为了避免这个情况，PLC 的工作速度就要快。所以，大型 PLC 的工作速度总是比小的要快。

控制规模还与内存区的大小有关。规模大，用户程序长，要求有更大的用户存储区。同时点数多，系统的存储器输入、输出的信号区（输入输出继电器区或称输入、输出映射区）也大。这个区大，相应的内部器件也要增多，这些都要求有更大的系统存储区。

控制规模还与输入、输出电路数量有关。如控制规模为 1024 点，那就得有 1024 条 I/O 电路。这些电路集成于 I/O 模块中，而每个模块有多少路的 I/O 点，总是有数的。所以，规模大，所使用的模块也多。

控制规模还与 PLC 指令系统有关。规模大的 PLC 指令条数多，指令的功能也强，才能应付对点数多的系统进行控制的需要。

控制规模是对 PLC 其它性能指标起着制约作用的指标；是高低档 PLC 区别之所在，也是 PLC 划分为微、小、中、大和特大型机的惟一依据。

PLC 发展的趋势是控制规模越来越大。早期 PLC 能控制几百、几千点规模就不小了。而今，控制规模已发展到几万、几十万点的开关量。另外，还可控制几千、几万路的模拟量。

如 OMRON 早期的 C2000H 型 PLC，为大型机结构，最多控制点数仅 4096 点。而后期的 CS1 型 PLC，结构为中型机，最多控制点数，就可达 5000 多点。

再如 CQM1 控制规模仅 200 多点，而改进后的 CQM1H，其控制规模翻了一番。可达 500 多点。

西门子 S5 机型，最多控制点数仅 1 万点左右，而今的 S7400 型 PLC，其 417CPU 控制开关量，可达 128k（DI）/128k（DO）点，控制模拟量，可达 8k（AI）/8k（AO）路。

由于 PLC 配置概念的更新，不少厂家的 PLC 可实现多 CPU 配置，即在同一机架可安装多个 CPU 单元，而每个 CPU 单元又可控制海量的点（路）数。再加上 PLC 网络技术的发展，PLC 的可控制规模已不受什么限制了。可以做到，要控制多少点开关量，就可控制多少点开关量；要控制多少路模拟量，就可控制多少路模拟量。用 PLC 实现对整个城市规模的设备或系统的控制与管理已是现实。

总之，控制规模在扩大，已是 PLC 发展的一个趋势。

但也应指出，在增大控制规模的同时，各厂家也注意开发微型 PLC，改善小型 PLC。微型 PLC 的控制点数越来越小，但可控制的输出的电流很大。可直接取代继电器控制系统。像这样能把自身的系统做小，是其它控制所作不到的。

一方面控制规模不断增大，另一方面控制规模又不断缩小，目的只有一个，那就是使 PLC 应用所覆盖领域更为广阔，名副其实地成为工业控制的主角。

10.1.3　组成模块在增多

PLC 的结构虽有箱体及模块式之分，但从本质上看，箱体也是模块，只是它集成了更多的功能。在此，不妨把 PLC 的模块组成当作所有 PLC 的结构性能。

这个性能含义是指某型号 PLC 具有多少种模块，各种模块都有什么规格，并各具什么特点。

一般讲，规模大的 PLC，档次高的 PLC 模块的种类也多，规格也多，反映它的特点的性能指标也高。相反，小型 PLC、档次低的 PLC 模块种类也少，规格也少，性能也低。

组成 PLC 的模块是 PLC 的硬件基础，只有弄清所选用的 PLC 都具有那些模块及其特点，才能正确选用模块，去组成一台完整的 PLC。

随着 PLC 在前进，PLC 的组成模块也在增多。

早期 PLC 模块的种类不太多，多数仅有开关量、模拟量输入、输出模块，高速计数模块等，而今则增多了很多，如温度模块，流量模块，模糊控制模块，PID 控制模块，回路控制器，秤重模块，凸轮模块，遥感模块，定位模块，运动模块…。可处理相应的模拟量、脉冲量。

此外，还出现了具有很强的信息处理能力的智能单元，如 ASCII 单元、BASIC 单元，以至于计算机（286 级）单元、硬盘单元、软驱单元…。这些模块的使用，将极大的增强 PLC 信息处理的功能。

不仅类型增多，性能也在提高。如，同样功能的模块，有的还带电隔离的，以提高抗干扰能力。有的还可实现故障自珍断，提高工作的可靠性。还有的可实现智能化，不是靠 CPU I/O 刷新，而是自身的 CPU，主动与内存交换数据。等等。

新式模块的不停涌现，组成模块不断增多，特别是智能单元的不断增加及应用，已使拥有这些模块的 PLC 的功能进一步增强。它已不再是原来意义的 PLC 了，而既是一个可在工业环境下使用的功能强大的控制机，且还是一个智能高超的信息处理机。

10.1.4　内存容量在增大

PLC 内存有用户及系统两大部分。用户内存主要用以存储用户程序。系统内存主要为用户提供程序数据区。

内存容量大，可存储的用户程序量也大，也就可进行更为复杂的控制。小型 PLC 的内存容量小。大型 PLC 的内存容量大，可达几十 k、几百 k。

从发展趋势看，内存容量是在不断增大着。如早期的 PLC，C60P，它的程序内存才 2k 字，而 CJ1 型 PLC，其最大程序的内存可达 120k 步（与字大体相当）。10 几年的发展，增大了几十倍。

系统内存提供的程序数据区，对于用户而言，主要体现在 PLC 能提供多少内部器件。不同的内部器件占据系统内存的不同区域。在物理上，并没有这些器件，仅仅为 RAM 或 EERAM。但通过运行程序进行使用时，给使用者提供的却实实在在有这些器件。

随着 PLC 的前进，这些器件不仅数量在增大，而且种类也在增多。

如 C60P，其 DM 区才 64B，而且在用它的高速计数功能时，其中的 32 个字还要归高速计数专用。而同样是小型机 CP1H 的 DM 区多 32k 字，增大了千倍。

同时，内部器件种类及数量的也在增多。

此外，内存的媒体、形式也多样化了。除了 RAM，EEROM，ROM，还有闪烁内存。除了基本内存，还有辅助内存，即内存卡。内存卡容量可达几十、几百 M。

内存卡的读写可用 PLC 指令管理。可从内存卡读取程序，并载入主内存，即读、即用。也可把主内存中的程序写入内存卡，写入后可长期保存。

内存卡不仅可读写用户程序，还可读写用户数据。所有的这些读写，都可用计算机文件的形式处理。如在计算机上，配置有专门接口卡，计算机也可直接对其读写这个文件。

内存还可带实时时钟，即使 PLC 不接电源，但由于有备份电池，内存中的实时时钟仍走时。

过去，为保持 RAM 内存中的程序、数据，用了支持电池。现有的用大电容，PLC 一个星期不工作，数据也不丢。

……

内存种类增多，容量增大，功能增强，越便于可以配备有这样内存的 PLC，进行控制与数据处理。显然，内存的进步是 PLC 在前进的重要标志。

10.1.5 指令系统在增强

PLC 有多少条指令，各条指令又具有什么功能，与 PLC 的功能密切相关，是 PLC 性能的重要方面。

PLC 每一次改型，其指令条数都在增加。现在小型机的指令已超过百条，大型机的功能指令，其代码已为三位数。

如 C60P 型 PLC，仅 37 条指令。功能很强的指令几乎没有。而 C60H 型 PLC，就有 100 多条指令，还有很多功能很强的指令。

再如同样是小型 PLC 的 CP1H，拥有的指令近千条，其功能码，用 3 位数代表。按类型计有 30 多种。不仅有位、字数据处理指令，还有字符、文件处理指令。普通计算机有的指令，它几乎也都有。

功能强的指令，如 MAX 指令，可在一组数据中求出它的最大值。若无此指令，仅用很多比较指令进行操作，那要麻烦得多。

功能复合的指令也不断增加。如指令的微分执行，实质就是微分指令及指令本身两个功能的复合。带感叹号的指令，带箭头的指令，则是输入、输出刷新及装载或输出指令的复合。指令多，功能强，又有复合功能，给编写出简明，可读性、可改性强，能执行复杂工作任务的程序提供了保证。

此外，由于新型 PLC 多增加有 PLC 厂家提供的功能块。实质上它也是指令，而且，功能比普通指令更强。同时，用户自己也可开发功能块。这也可认为是自己生成的指令。这样，也可以这么说，PLC 的指令集是无限的。要多少就有多少。

正是 PLC 指令系统日益增强。进而使它的应用日益扩大。PLC 成为工业控制的一大支柱，与它具有功能强大的指令系统是不无关系的。

10.1.6 工作可靠性在提高

可靠措施的目的是增加 PLC 平均故障间隔时间（Mean Time Between Failure，MTBF）及

减少 PLC 的平均修复时间（Mean Time To Repair，MTTR），以提高 PLC 的有效度 A（Availability）。

$$A = MTBF/(MTBF + MTTR)$$

式中　A ——有效度；

MTBF——平均故障间隔时间；

MTTR——平均修复时间。

当然，这里的 A 值越大越好，它可使 PLC 系统得到充分的利用。而从上式可知，MTBF 越大，MTTR 越小，则 A 越大。所以，PLC 的可靠措施都是围绕提高 MTBF 及降低 MTTR 值进行的。

鉴于可靠工作是 PLC 的重要特点，故有关提高 MTBF 及降低 MTTR 的措施如何，以及 PLC 的 MTBF 与 MTTR 值也成为 PLC 性能的重要指标。

PLC 本来就很可靠，这是它的基本特点之一。随着技术的发展，它的集成度越来越高，内部器件越来越少，并随着制造工艺的进步，其可靠性也更进一步提高。当前，PLC 多不是用坏，而是由于使用不当才出现故障的，或由于技术进步而被淘汰。

另外，为了做到万无一失，不少厂家还推出种种冗余或容错配置的解决方案，更极大地提高了它的工作可靠性。

目前，这个冗余或容错配置，不仅可在 CPU 模块实现，也可在电源模块、I/O 模块、回路控制模块实现。不仅可在当地实现，也可在网络上实现。

有了这些，即使使用了复杂的系统配置，也可加大 PLC 的平均故障间隔时间、MTBF（Mean Time Between Failure）。目前，此时间最长的可达到近 70 万小时。而早期的产品才 5 万多小时。

同时，有的厂家还开发有一系列专门技术以提高可靠性。如西门子的接地、抗干扰有专门措施，GE FANUC 对输入短路或断线也有监测的技术，等等。再就是，各厂家在 PLC 自诊断，出错记录，以及对系统的监视，都有很多措施。

这使得 PLC 即使出了故障，由于有记录可查，也很易排故。再加上，如 PLC 为模块化配置时，出现故障更换有故障的模块即可。而且，有的冗余或容错的配置的 PLC，可在线更换模块。

PLC 工作可靠的进步，还表现在 PLC 维护更为简单，以至于可以零维护。如当今的 PLC 都无须风扇，不少模块能带电即插即用，内存不需电池支持，等等。而这些，早期的 PLC 都是没有的。

显然，有了这些，可大大减少 PLC 的平均修复时间、MTTR（Mean Time To Repair）。

此外，可靠措施还体现可在特殊环境下使用的特殊 PLC 问世。这将在 PLC 类型在增加一节中再具体介绍。

10.1.7　联网能力在增强

联网通信，主要是指 PLC 与计算机，与其它 PLC，与智能设备联网通信。联网通信的方法又多、又灵活，说明联网通信能力强。

联网能力还表现在网络规模（联网节点的多少）、网络覆盖范围（数据传送距离）、网络连接介质、网络的互联及网络的兼容性上。目前，有多到几十、几百个节点，覆盖范围大

到达几十、几百 km。使用的介质有双绞线、同轴电缆、光缆以及无线介质。多个网络互连，不同厂家的 PLC 共在一个网络上，都已成为现实了。

联网能力还表现在传送数据帧的大小及数据传送速率上。可传送的数据帧当然越大越好，传送速率当然越高越好。目前，PLC 数据传送帧，已从几十字节发展到几百、几千个字节，传送速率也已从几千发展到几万、几十万，几百万位每秒，甚至出现了百、千 M 级的工业以太网。

联网能力还表现在网络的开放性上。现在已出现了如 DeviceNet、Profibus、CC-Link 等由几个 PLC 大厂家支撑的 PLC 网络，具有相当规模的开放性。多数 PLC 可与这些网络互连。

联网能力还表现在 PLC 与工业以太网以至于互联网相连上。进入 21 世纪，PLC 的这方面的进步，使 PLC 控制信息化、远程化又推进了一大步。

联网，扩大控制规模与能力，形成广域与远程控制，已是 PLC 发展的重要特点。PLC 再也不是控制系统的“信息孤岛”或“控制孤岛”了。PLC 这些网络性能的提高，功能的增强，也说明 PLC 在前进。

10.1.8 外部设备在丰富

除了种种模块，PLC 还有很多外部设备。发展也很快，且越来越丰富。尽管用 PLC 实现对系统的控制可不用外部设备，配置好合适的模块就行了。然而，要对 PLC 编程，要监控 PLC 及其所控制的系统的工作状况，以及存储用户程序、打印数据等，就得使用 PLC 的外部设备。故一种 PLC 的性能如何，与这种 PLC 所具外部设备丰富与否，外部设备好用与否直接相关。

PLC 的外部设备有四大类：编程设备、监控设备、存储设备、输入输出设备。

编程器这几年没有什么发展，因为用计算机加编程软件编程，比用编程器编程要方便得多。但监控设备发展是很快的。目前，已拥有大、中、小型配套的，高、中、低档都有的人机界面，为对 PLC 的数据监控提供了很大的方便。PLC 界面不太友好的状况将成为历史。目前的问题只是，这些人机界面的价格还是稍高，无法普及。

作为外存储设备，过去拥有磁带、软磁盘，也没有发展。主要是闪存的发展，完全可用内存卡代替它。

输入输出设备发展也很快。输入设备的有条码读入器，输入模拟量的电位器等。输出的有打印机。随着技术进步，这种设备将更加丰富。

外部设备已发展成为 PLC 系统的不可分割的一个部分。它的发展与进步是 PLC 必须在前进的重要内容。

应指出是，既可作为控制器，又要作为数据处理器的 PLC，其外部设备还有待发展。特别要开发一些使用方便，价格低廉，工作可靠，即插即用的可视化外部设备，以便于建立友好的，类似 DCS 操作站那样的界面，供操作者使用。

10.1.9 支持软件在完善

为了便于编写 PLC 程序，多数 PLC 厂商都开发了相应的编程支持软件，为 PLC 运用多种编程语言进行编程、监控提供平台。同时它还是 PLC 硬件组态或软设定的工具。

为了用好各种高功能的智能硬件模块，PLC 厂商都开发了与其配套配置和使用软件。所

以，这类软件好用与否，也是 PLC 支持软件好坏的一个指标。

应该讲，早期 PLC 并没有什么编程软件，编程用的是编程器，不用计算机。OMRON 第一个编程软件叫 LSS（梯形图 LADER，支持 SUPPORT，软件 SOFTWARE）出现在上世纪 90 年代初期。当时计算机的操作系统为 DOS，所以，软件的界面不好，也很不完善。

后来，OMRON 推出中文界面的 SSS 软件，但也只能运行在 DOS 的操作系统上。

第 1 个 WINDOWS 界面的软件为上世纪末推出的 CPT，在新旧世纪交替过程中，又推出 CXP。其版本已从 1.0 发展到 4.X 甚至更高。近几年又把 OMRON 所有的控制与监控软件集成为软件套装，即 CX-one，也更新了多个版本。其功能、界面都更加完善。而且，与别的厂家的新软件不同，它能与过去的软件兼容。可读取用旧软件编的程序，并可进行转换。说明 OMRON 用心良苦，已把编程软件发展到相当完善的地步。

总之，为了用好 PLC，PLC 的支持软件越来越丰富，性能也越来越好，其界面也越来越友好，也因此，它的情况如何，也已成为评判 PLC 性能的指标之一。它的进步，也是 PLC 在前进的又一侧面。

但也应指出，由于 PLC 系统越来越复杂，PLC 还应有更为方便的编程、图形监控及组态软件。最好能有像 DCS 工程师站所配备的软件那样，很便于编程、监控与组态。

10.1.10　经济效益在增加

以上九条讲的都是 PLC 的技术性能。其实，使用 PLC，还要考虑经济指标。“经济是基础”，经济上不合算，不能带来经济效益，使用 PLC 也就没有基础。所以，这个指标的进步也是重要的。

经济效益最简单的就是用产出投入比。而这个比还与 PLC 自身的性能价格比有关。性能体现产出，价格反映投入。这个比大，效益自然也就高。

PLC 的功能、性能一代胜过一代。但它的价格反而低了。如，同样点数的 P 型 PLC，与 CP1H 相比，后者反而便宜，但功能、性能却高过千倍。

PLC 的性能价格比越来越提高，为 PLC 的广泛应用，提供了强有力的经济基础。因为归根到底，人们使用 PLC 这个新技术，就是它能带来经济效益，而性价比的提高，则使这个获益成为可能。

也许，这也是 PLC 在前进的一个重要表现。如果不是这样，而是随着 PLC 性能的提升，其价格越来越高，那就不妙了，就不能算 PLC 在前进了。

10.2　PLC 的应用在扩展

关键词：自动化、远程化、信息化、智能化

PLC 的应用正在不断地扩展着。不仅在顺序控制上大显身手，而且在过程控制、运动控制上也越来越发挥着重要作用。PLC 已成为当今系统工作自动化最有效的工具之一。

同时，PLC 在系统控制的远程化、信息化及智能化上也是大有作为的，与别的手段相比，也是最合算的。

PLC 应用这样不断地的扩展，恰好说明 PLC 在前进。

10.2.1　PLC 用于系统控制自动化

系统控制自动化是指，不用或很少用人的干预，系统能自行工作、实现系统自身的功能。这是人们设计与使用系统所追求的目标，也越来越多的变成了现实。而自动化则要通过有效的控制才能实现。这有效控制有：顺序控制、过程控制及运动控制。

1. 顺序控制

顺序控制是系统工作最基本的控制，也是离散生产过程最常用的控制。

顺序控制是 PLC 的初衷，也是 PLC 的强项。在顺序控制领域，至今还没有别的控制器能够取代。

2. 过程控制

以前，过程控制多使用集散系统。而今使用 PLC 已是一个趋势。因为 PLC 价格低，在实施过程控制的同时，还可进行其它控制。再加上新的过程控制模块的开发，以及相关软件的推出，用 PLC 进行过程控制已变得很容易。所以，目前有的厂家 PLC 用于过程控制的产值份额，已超过用于顺序控制。

3. 顺序控制

PLC 用于运动控制，价格低，在进行运动控制的同时，还可进行其它控制。再加上新的运动控制模块的开发，以及相关软件的推出，用 PLC 进行运动控制已变得很容易。所以，目前有的厂家 PLC 用于运动控制的产值份额与日俱增。在相当程度上，可以代替价格比其昂贵的数控系统。

10.2.2　PLC 用于系统控制远程化

系统控制远程化是指对系统的远程部分的行为及其效果实施检测与控制。它既是自动化发展的结果，也是一些现实系统（如不宜人们接近的系统，或像自来水公司那样，本来就是分布工作的系统）控制的需要。

远程化对提高控制效益也是重要的。如山东威海市自来水公司，用 PLC 对水资源进行远程化管理与控制，就可做到使该城市有限的水资源得到有效的利用，避免了浪费，保证了居民用水的基本需求。

实现了远程化可从系统的各个角落收集相关信息。可保证信息的完整。为信息的使用创造的前提。

远程化的基础在联网和通信技术。目前，PLC 基本上已完全实现网络化，所以 PLC 用于系统工作远程化已是就便之举。

在系统控制远程化中，PLC 处于最关键的位置。向下，它连接设备网，对现场设备实施控制，或从中采集数据。向上，它连接信息网，可向上位机传送数据，或接受控制信息。在中部，还可连接控制网，与其它 PLC 交换数据。在此，PLC 扮演着承上启下、左右贯通的重要角色。

总之，PLC 可在系统控制远程化中发挥重要作用，而系统控制远程化也离不开 PLC 的具体参与。

10.2.3　PLC 用于系统控制信息化

系统控制信息化是指，系统的行为与行为效果量（数字）化、检测，采集、处理、存

储、显示与传送。一句话，信息化就是用数字表示系统的状态。数字地球、数字奥运，不能不说其含义也许与此有相同之处。

用 PLC 对已量化了的系统的行为与行为效果，进行检测，采集、处理、存储、显示与传送是很方便的。特别是数据采集，它的及时性、准确性及完整性，则是实现信息化的基础。

信息化目的是使系统透明，人们要了解什么，就可得到什么。法国施耐德公司提出透明工厂（Transparent Factory）的概念，也是便于人们对工厂信息方便了解。

有了信息化，可把工厂的生产过程自动化与生产的原料管理、产品销售结合，作到供、产、销无缝连接。进而实现按市场需求生产，按生产需要进料。这样的工厂，可运用零库存管理。工厂的资源、资金将得到充分利用，其效益当然会是很高的。

自动化与信息化结合，或说在信息化基础上的实现自动化，是当代自动化与传统自动化的重大区别。当然，当代自动化与传统自动化的另一区别是，在实现自动化的手段上，使用了信息技术。

10.2.4　PLC 用于系统控制智能化

系统控制智能化是指，使系统具有一定的智力，能初始化系统，能记录、判断、应对非正常情况，或能对非正常情况的处理提供必要的信息。如能自整定系统工作参数、自调整系统工况、自调整工作模式、自诊断、自修复等等。

随着系统自动化、远程化、信息化的推进，系统也越来越复杂。系统的使用、维护也变得越来越困难。这时，如果系统不是智能的，没有一定的智力，既不便于系统的使用，又难以发挥系统效能。极端的说，一个没有智能的复杂系统，出现故障，维修之难是可以想象的。所以，系统控制智能化是更好地实现系统控制自动化、远程化及信息化的要求，也是系统控制自动化、远程化及信息化技术进步的必然趋势。

用 PLC 实现系统控制智能化要做的工作还很多，要走的路程还很远。也可说这个任务目前仅仅是开始。相信随着 PLC 应用的普及与提高，这个目标是会实现的。

应指出的是，系统控制自动化、远程化、信息化及智能化是相对的，也是分层次的。因而也是发展、变化的。

例如，自动化，多大范围实现自动化，一个设备，一个生产线，还是一个车间、工厂？远程化，拥有多少节点，覆盖多大地域？信息化，采集与传送信息可多可少，这都是动态的。智能化，也有个怎么逐步推进的问题，等等。这些，只能是逐步发展与不断完善的。相信随着 PLC 技术的进步，越来越完善地实现自动化、远程化、信息化及智能化的 PLC 控制的系统，是会越来越多的。

10.3　PLC 的概念在更新

关键词：工作模式、系统结构、Producer/Consumer、多 CPU 配置、软设定、内冗余、智能工厂、e自动化、全集成系统、透明工厂

PLC 性能在提高，PLC 应用在扩展，来自 PLC 的概念在更新。无论它的工作模式、系统结构、设定手段，还是编程方法、可靠设计及追求目标，都有很大变化。有不少思想创

新。“思想是行动的先导”，这些思想创新为 PLC 的发展，注入的生机，提供了动力，开辟了新路，指明了方向。

以下将对今年 PLC 主要的概念更新，或说思想创新，作些简要说明。

10.3.1 工作模式

工作模式是指 PLC 操作系统对程序及 I/O、通信的管理方法及方案。在这方面，PLC 有很多创新。

传统的 PLC 工作模式，基本上是扫描工作模式。用户程序周而复始的运行，再加输入、输出刷新、通信服务、外设服务及系统自检。后来出现扫描加中断。而今在好多方面已有所突破：

（1）并行处理。CJ1H 型 PLC 可实现平行处理，程序执行与外设服务可同时进行，既提高程序响应速度，又可说缩短通信时间。CJ1H 型 PLC 可选用两种并行处理模式。一种是在程序执行周期中用同步存储器访问方式刷新 I/O 的并行处理；另一种是在外围服务周期中用异步存储器访问方式刷新 I/O 存储器的并行处理。

（2）指令可多周期执行。CJ 系列 PLC 有很多功能很强的数据、文字处理指令，如排序，求平均值等。如在一个扫描周期中完成指令的功能，所占的时间太长，势必导致扫描周期的不均。为此，CJ 系列 PLC 有的指令允许在多个周期中执行，以到实现指令的功能。

（3）多任务组织。贝克莱 PLC 的操作系统，把程序按多任务组织。不同的任务设定不同的扫描间隔及时间片。紧急的、重要的扫描间隔短，时间片长。让其实现功有充分的保证。其它的扫描间隔长，时间片短。实现其功能可能要用多个较长时间。此操作系统类似多任务的计算机操作系统。贝克莱称，此 PLC 应叫 PCC，即可编程计算机控制器，而不是可编程逻辑控制器。

（4）I/O 刷新有所突破。传统的执行程序前后进行。后发展为可立即刷新。接着，又出现带复合功能的指令执行，如带感叹号的装载指令。可做到先刷新，后执行。还有带感叹号的输出指令。可做到执行后就刷新。这些都极大地提高 PLC 对重要信号的响应速度。

CV 型 PLC 除了循环（周期）刷新，完成一个循环，即对所有的 I/O 进行刷新。还可定时刷新，如定 10～100ms 刷新全部的 I/O。还有“过零”刷新，当交变信号过零时进行刷新，便于改善交流信号传送的条件。等等。

更值得提及的是 AB 公司的新型 I/O 模块。自带有 CPU。它的 I/O 刷新不是被动实现而是主动进行的。如输入模块称之为数据生产者（producer），它可定时的，或在数据变化时，把数据送给输入映射区或链接区。再如输出模块，称之为数据消费者（consumer），它可定时把数据送从输出映射区或链接区，读取数据。

10.3.2 系统结构

系统结构是指，PLC 的组成原则及组成方案。

上世纪末问世的 OMRON CV 型 PLC，其系统结构就有不少创新。如采用了统一的总线结构和多微处理器的设计。第一条总线是 I/O 总线，用于进行 CPU 与一般的 I/O 通信。第二条总线是 CPU 总线，使得无需 CPU 控制，也可在 CPU、协处理器及通信模块等 CPU 总线单元之间进行高速的、点对点的总线通信。这种结构不但方便了通信，而且可使执行程序及

通信处理分开，减少 PLC 的扫描（CV 机称循环）时间。

这种结构有助于 CV 型 PLC 实现三 V。这三 V 是英文 Visual（可视性）、Variable（可变性）及 Valuable（可带来附加值）的首字母。

可视性是指可在 CV 型 PLC 的任何机架上安装编程器，监视 CV 型 PLC 的工作情况及读出所需要的数据。

可变性是指 CV 型 PLC 在运行中可设定数据，以至程序改变，非常灵活。

可带来附加值是指 CV 型 PLC 的使用，除本身的价值，还可带来附加的价值。CV 机基本上实现了这三个 V，因此才在 C 之后加上 V 而得名。

PLC 系统结构创新还体现在以下几个方面：

（1）从箱体走向模块，模块从有底板到出现无底板，又出现内装式的 PLC，结构越来越新颖。

（2）从单一品种走向家族化，每一个型别，都有多种规格，选用方便。

（3）从集中走向分布。

（4）从单一 CPU 模块走可多 CPU 模块。最多的，一个机架可安装 4 个 CPU 模块。图 10-1 所示为 S7-400 多 CPU 配置的例子。

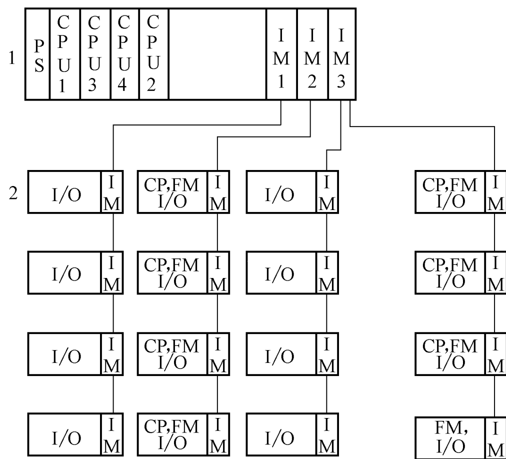


图 10-1　多 CPU 系统
1—CR　2—ER

从图知，它的主机架上配备有 4 个 CPU。这 4 个 CPU 相互通信，同步自动工作，但各独立执行各的程序，各与各的信号模块（SM）、功能模块（FM）、通信模块（CP）及 I/O 模块相连。故控制任务可并行处理。

而什么时候需要多 CPU 模块呢？程序量太大，一个 CPU 与内存难以处理，用多 CPU 分担处理；或系统中个别要求用特别快的速度处理，可另配置相应的 CPU。如果任务分工明确，用多 CPU 也好管理。

（5）从封闭走向开放，自身系统也许别的公司的模块接入，或用自身模块接入别的公司的系统。还有专门接口模块，用于不同系统对接。

（6）软件 PLC 的出现，显然是系统结构思想创新的一个成果。

10.3.3　设定手段

PLC 设定手段也有很大变化。

早期，PLC 设定多是用硬件实现。具体是用跳线，用 DIP 开关，用旋转开关。如用以与上位计算机链接、通信的 C200H 的 Host Link 单元就是用多个旋转开关和 DIP 开关进行波特率、数据位、奇偶校验、命令等级等通信参数的设定的。

硬件设定是很不方便的。每次设定改变，都要停机，切断电源，甚至模块还要从机架卸下，设定后还要再装上。

硬件设定的信息量少。要作很多设定时，都用设定开关，那要用很多。PLC 体积又那么小，往哪儿放？

随着存储硬件技术的发展，人们发现用 EERAM，后来用闪存存储数据，掉电时也不丢失。用其存储设定设定数据，是很方便的。所占的体积小，信息量大，工艺又简单。所以，很快就发展起来了。这种设定称软设定，像计算机 BOIS 那样，可保存大量的信息。

最早提出全部用软设定代替硬件设定的是美国 GE 公司。当它推出 90-70 型 PLC 时，曾说“Good bye hard setup!”（再见，硬设定 !）。所以，他的 90 系列 PLC 就没有硬设定，只用软设定。

从硬设定改变为软设定，是当今一个趋势。OMRON 小型 PLC 用了几十个 DM 字进行有关设定，也含以上讲的通信参数设定。这也是这个趋势的体现。

但软设定也有不足。得记住很多“重要”的设定值。如 OMRON 小型 PLC 程序写保护，可把 DM6602 字的 03 ~00 位设为 1。作了这样设定后，如忘了，那再改写程序就不可能了。为此，也给不少用户带来过麻烦。

这个进一步发展是，用工具软件的设定窗口设定，代替手工更改设定字的值。这么做，从根本上讲，当然还是改写设定字，但它用工具软件实现。所以，不用记住这些值，而且，设定、更改都很方便。

PLC 设定的进步还表现在以下几个方面：

（1）机架、模块的编号或说地址，不用固定编号、固定地址，而可由用户编号，用户编址。

（2）内部器件的数量，不是固定的，用户可按需选定。少用内部器件省下的内存，可另作它用。器件的编号也可不固定。也可由用户编排。

（3）设定值的改变，尽量做到即改即实现。而不是改后还得重新上电才能实现。有一办法是用模块起动位。一个要设定的模块，都有与其地址对应的起动位。一旦这个位 ON，这个模块新设定就可生效。新设定生效后，自动又使这个位复位。这比重新上电要方便多了。最好连这一步都不要，即改即实现，可动态设定。

（4）PLC 的操作系统可升级。出现了新模块，升级后的操作系统也可支持。这类似计算机主板的 BIOS，大容量的硬盘不支持，但升级后就支持了。

……

当然，以上这些进步，只是个别厂家、个别型别的 PLC 才有的。但相信将会有越来越多的 PLC 跟上来的。

此外，AB、施耐德等高档 PLC 不仅设定软件化，同时操作系统的版本也可以升级，而且可以远程操作实现。这为用户现有 PLC 的“与时俱进”，享用厂家的新技术成果提供了可能。

总之，PLC 设定技术的进步，为使用 PLC，更新 PLC 都提供了更大的方便。

10.3.4 编程方法

PLC 编程工具、编程语言、编程方法、程序组织发展也很快。也有很多思想创新。

编程工具已很少使用编程器，而用计算机加可视化编程软件。

编程语言也增加了多种。不再仅是助记符、梯形图。在 IEC61113 标准中，还规定有功能块、顺序流程图、语句表语言。现多数 PLC 也都可使用这类语言。

此外，还出现一些近乎自然语言，可为不懂得电器技术，但又是设备使用专家所认识的

语言，如 OMRON F 系列 PLC 使用的系统流程图就是此类语言。

与编程语言相应的编程方法也有不少思想创新。已由线性编程（所有程序集中在一起），发展到分块编程、分块管理。OMRON 出现有段（Section）、任务，为程序的编写、修改、重用都提供了很大方便。

还可设想将有真正面向对象的语言及更多的可视化的方法问世。能使 PLC 的指令组件化，组件的接口标准化。选择组件，组件连线及填写接口参数，就相当于编程。果能如此，编程方法将又是一次飞跃。

在上世纪 90 年代初期，美国 IPM 公司的 IPM1612 型 PLC，曾有用计算机仿真编程。不用真 PLC 也可调试程序。如今，这类仿真软件也已很多了。且界面很好，仿真的项目多，很有真实感。仿真软件既可连续运行程序，也可单步调试，对学习编程、简化程序调试都有很大帮助。这也是 PLC 编程方法的创新之一。

除了仿真，联机调试也有创新。除了可对输入点强制，人工的设定输入状态，有的 PLC 在联机时，也可像调 VB、VC 程序那样，单步运行程序。为查找程序错误提供很大方便。

再就是，编程中导入、导出也用得很多。如 CXP 的符号表设计，可用 EXCEL 进行。设计好了，再导入到 CXP 中去。尽量使专用软件设计，也可用通用的软件及方法实现。

10.3.5　可靠性设计

PLC 工作是很可靠的。尽管如此，PLC 的可靠性设计概念也还在更新。

（1）冗余在扩大，从 CPU 冗余，扩展到电源冗余、I/O 冗余、网络冗余、特殊模块冗余等等。

（2）允许模块在线插拔，可在运行中进行系统重组。

（3）从冗余发展到容错，可带故障运行。加上（2）的设想，也可在线排故。

（4）在冗余、容错的同时，加强安全保证措施。避免一旦出现故障，也要确保人身安全及设备安全。

（5）从用外部冗余、容错，即多个 CPU、多个模块，发展到用内部冗余、容错，即 CPU 模块内部处理器冗余、I/O 模块内 I/O 可靠性设计。这既增加了可靠性，又减小了体积、降低了费用。为 PLC 走向更多的冗余设计提供可能。

10.3.6　追求目标

早期 PLC 追求的目标仅仅是取代继电器控制，能“控制各种机械或工作程序”、“控制各种类型的机械或生产过程”。看看 GM 的 10 条就很清楚了。

如今，PLC 追求的目标是实现系统的自动化、远程化、信息化及智能化。各 PLC 厂家都有各自的具体目标：

1. 智能工厂（Smart Factory）

这是 OMRON 公司在上世纪末，提出的目标。目的是通过 OMRON 产品的开发与进步，能使工厂的自动化有所推进。能从传统的单机自动化，即点上的自动化，发展到线，即生产线的自动化；进而再发展到面，即整个车间、以至于工厂，也能实现自动化。而且还有智能化的要求。

也因此，在多年来，OMRON 产品也为实现此目标在前进着。CV 型、CS 型、CJ 型 PLC

的相继推出，网络技术的长足进步，配套软件的极大改进，人机界面发展与性能大幅度提高。难道不正是为实现这个目标迈出的一个个坚实的前进步伐吗！

2. e 自动化（eAutomation）

这是美国 AB 公司近期提出的目标。强调了用 PLC 实现自动化，要以信息化为基础。这里的“e”实质是信息化。

它的目的是解决由于信息不畅，工厂的供、产、销老是衔接不好的问题。所以，它的信息化基础上的自动化，要求工厂的供、产、销，做到无缝（Seamless Connect）衔接。果真如此，那对提高工厂的资金利用及经济效益将是很大的推进。

AB 公司的 e 自动化还有个目标就是工厂设备管理要信息化、智能化。传统的设备管理是坏了才修，这当然风险很大，也很容易造成工厂管理的被动。后来发展为计划预修。生产不会出现被动。但有时没有坏也修，也是浪费。如今 AB 提出基于故障检测的维修（Base Fail Detect Maintenance and Repair），对重要设备，用 PLC 在对其控制，实现其功能的同时，还对其跟踪诊断，建立信息系统。这样就可做到：设备要坏但还未坏的时候安排修理。就可做到既不浪费修理费，又不耽误生产。

当然，这些目标的实现也是靠它的 PLC 技术的发展。AB 公司近期推出的种种 PLC 的新品牌、新机型，特别是网络技术、软件技术的极大进步，也正是为此而努力的结果。

3. 全集成系统（Total Integrate System）

这是德国西门子公司提出的目标。随着自动化技术的不断发展和计算机技术的飞速进步，今天的自动化控制概念也发生了巨大的变化。在传统的自动化解决方案中，自动化控制实际上是由各种独立的、分离的技术和不同厂家的产品来搭配起来的，比如一个大型工厂经常是由过程控制系统、可编程控制器、上位监控计算机、SCADA 系统和人机界面产品共同进行控制。为了把所有这些产品组合在一起，需要采用各种类型和不同厂商的接口软件和硬件来连接、配置和调试。

而全集成自动化的思想就是，用一种系统完成原来由多种系统搭配起来才能完成的所有功能。应用这种解决方案可以大大简化系统的结构，减少了大量接口部件，应用全集成自动化，可以消除上位机和工业控制器之间，连续控制和逻辑控制之间，集中与分散之间的界限。

同时，全集成自动化解决方案还可以为所有的自动化应用提供统一的技术环境，这主要包括：统一的数据管理，统一的通信 ，统一的组态和编程软件。

基于这种环境，各种各样不同的技术可以在一个用户接口下，集成在一个有全局数据库的总体平台中，这样系统之间的接口费用大大降低，备品备件的品种和数量也大大减少。同时技术人员可以在一个平台下，对所有应用进行组态和编程和监控，可以大大提高监控水平，减少非计划停车时间。

4. 透明工厂（Transparent Factory）

这是法国施耐德公司提出的目标。施耐德的 Quantum FactoryCast 嵌入式 Web 服务器和 Quantum 以太网 I/O 扫描服务器模板，可立即将 Quantum PLC 接入以太网，并使之变成具有友好的 Internet 界面的控制器。

Quantum FactoryCast 嵌入式 Web 服务器，带有预配置的 FactoryCast 2.0 软件。网络模板提供 8M 存储空间，用于下装用户自定义的 HTML 网页和其它出厂预装的网页工具。

预装的工具包含一个完整的 JavaBean 环境。用户可通过使用 Schneider 提供的预配置的 JavaBean 库的网页模板，自定义可以动态显示控制器实时数据的页面。网页模板非常容易使用，不需要 Java 的编程经验。页面可使用标准的浏览器环境，如 Netscape Navigator 或 MS Internet Explorer。

一旦设置完成，对于任何可以通过以太网和浏览器与之连接的客户，网络模板就成为网页服务器（当然带有口令安全保护功能!）。这样，PLC 的数据，从工厂内部到世界各地都非常容易地访问（当然需要通过必要的以太网/互联网连接和安全目的的防火墙等）。

Quantum Ethernet I/O 扫描服务器，可用于读写与 Quantum 相连的、循环扫描的多达 64 个设备的大量数据，还可以作为服务器支持 32 个同时通过以太网连接的 Modbus-TCP/IP 设备。如果正确地使用交换式以太网，该模板可提供极其快速的数据采集及工厂级网络控制（每秒可处理 4000 个以上 Modbus TCP 报文，4000 × 125 = 500000 个寄存器字）。模板还可以利用内置的 BootP 服务器，自动地为包括 Momentum 在内的各种 I/O 设定 IP 地址。所有的配置管理表格只需要通过普通浏览器即可实现。

以上讲的都是国外 PLC 厂家追求的目标。我们国内 PLC 厂家也有自己的目标，也将进一步发展、推进自己的产品，实现自己的目标。进而为中国，以至于世界的第一产业、第二产业、第三产业的自动化、远程化、信息化及智能化作出自己应有的贡献。

10.4　PLC 的类型在增加

关键词：环境条件扩展、微型 PLC、分布式 PLC、内装式PLC、安全型 PLC、运动控制PLC、过程控制PLC、软件PLC

思想创新，观念更新，以及 PLC 应用的扩展，导致越来越多新类型的 PLC 问世。以下将对这些作简要介绍。

10.4.1　环境条件扩展型 PLC

PLC 的工作与保存环境一直是有一定要求的。以 OMRON PLC 为例，其存储温度不能低于摄氏零下 10 度，也不能高于摄氏零上 60 度，其工作温度不能低于摄氏 0 度，也不能高于摄氏零上 60 度。等等。

但有些 PLC 要在野外工作。如高速公路的路灯控制，油田的一些设备控制，车载设备控制，等等，在北方地区工作时，其工作温度要低于零下。有的在高温环境下工作，其环境温度也很高。

所以，能突破这个环境条件限制的 PLC 相继出现了。如施耐德公司的 TSX Compact 型 PLC，就是一个环境条件扩展型的 PLC。它结构小巧，功能强大，存储器、处理速度、I/O，环境指标、编程软件等方面性能都较高。从而使其成为众多控制和 RTU 应用的更完善，更灵活的解决方案。

其具体性能指标为：

CPU 单元：386 的控制器，支持 I/O 字容量 128-512。

I/O 方式：通过 InterBus 可以支持远程 I/O，包括 Compact 和 Momentum I/O。在 Quantum InterBus 网络上也可作为从方式。

编程：除支持IEC编程语言外，系统还支持改进后的984指令表。

工作温度：允许从 -40～70℃。

敷涂层保护：可选。

TSX Compact能够为众多应用提供强大，灵活而高度兼容的方案，如：汽车存放，包装机，压缩机和泵站控制，输送机，数据采集，变电站自动控制，能源管理，机械组装，油田和管线机械，压力机控制，铁道，冲压，水处理，电缆绕线机械等。

再如西门子的S7-300型PLC，已有环境条件扩展型的CPU模块，信号模块，接口模块和ET 200M分布式I/O模块。所配置的系统，允许工作的温度范围，可从 -25～+60℃，也适用于存在凝露，结冰和95%湿度的环境，并有更强的耐受振动和污染特性。

这种PLC系统非常适宜用于运输，水厂，石化行业和食品工业。适用于所有无空调的车间等，如火车，信号站和汇接站，水泥搅拌机，垃圾车，升降平台，桥梁悬臂，交通控制系统，污水处理厂和炼油厂，以及饮用水供应和管道用的泵站，冷藏库，冷藏货车和冷藏车。

预计，还将有更多的PLC厂家推出此类产品。

10.4.2　微型PLC

为了小型设备的控制需要，近年，不少PLC厂商开发了输入、输出点数很少，体积很小，价格低廉，输出电流很大（如8A以至于更大），可直接驱动执行器，并自带有简易编程装置的微型PLC。用其可代替工业继电器控制系统。因此，有的厂家称之为可编程逻辑继电器，P（Programmer）L（Logic）R（Relay）。这些产品有：

1. OMRON ZEN系列PLC

除了主机，它还有扩展模块，可按需要选用。最多可扩展3个模块，输出全部为继电器，输出电流可达8A。

有关模块的主要性能如表10-1所示。

表10-1　ZEN主要模块特性

名称	类型	I/O	电源	输入	输入电压	输出	键液晶屏	内置时钟	模拟量输入	型　号
CPU	LCD	10	AC100～240V	6	AC100～240V	4	有	有	无	ZEN-10C1AR-A
	LED						无	无	无	ZEN-10C2AR-A
	LCD		DC24V	4	DC24V	4	有	有	有	ZEN-10C1DR-D
	LED						无	无	有	ZEN-10C2DR-D
扩展I/O		8	—	4	AC100～240V	4	—	—	—	ZEN-8EAR
			—	4	DC24V	4	—	—	—	ZEN-8EDR
		4	—	4	AC100～240V	—	—	—	—	ZEN-4EA
			—	4	DC24V	—	—	—	—	ZEN-4ED
			—	—	—	4	—	—	—	ZEN-4ER

2. 西门子logo

LOGO！基本型不同型号的主要性能如表10-2所示。

表 10-2　LOGO！基本型不同型号的主要性能

技术数据	LOGO!12RC	LOGO!24	LOGO!24RC	LOGO!230RC
输入	6	6	6	6
输入电压	DC12V	DC24V	AC/DC24V	AC115V,230V
输出	4 继电器	4 晶体管	4 继电器	4 继电器
连续电流	10A 对应阻性负载 3A 对应感性负载	0.3A	10A 对应阻性负载 3A 对应感性负载	10A 对应阻性负载 3A 对应感性负载
开关频率	2Hz 对应阻性负载 0.5Hz 对应感性负载	10Hz	2Hz 对应阻性负载 0.5Hz 对应感性负载	2Hz 对应阻性负载 0.5Hz 对应感性负载
内置时钟	8/典型 80h	—	8/典型 80h	8/典型 80h

3. 三菱公司 alpha 型 PLC

它的尺寸小，6、0 及 20 个 I/O 点，继电器输出电流可达 10A，也可选用半导体输出点，所用电源可选直流或交流，也有模拟量输入选件，有的有实时时钟并用 4 位数显示年，对 20 点的机型还配置有 AS-i 网络接口。另外，它还有仿真软件，可不需硬件即可调试程序。

10.4.3　分布式 PLC

分布式 PLC 是基于网络的。所控制的 I/O 点分布在各处，在当地可以没有 I/O 单元，不控制任何 I/O 点。OMRON 的 SRM1 机即为这样 PLC。图 10-2 所示为它的两个机型的 CPU 单元。

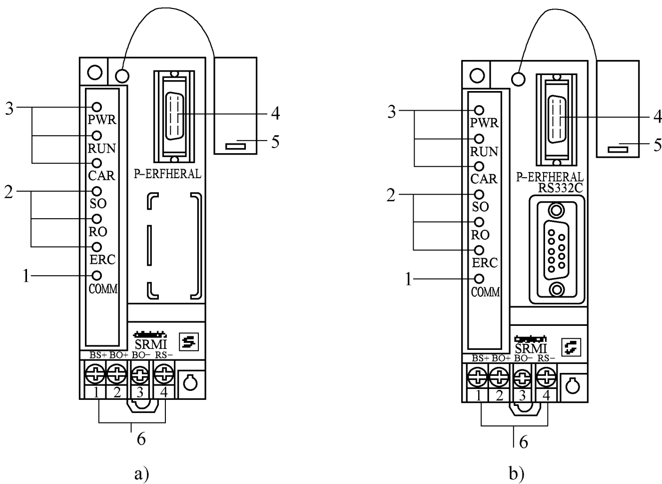


图 10-2　SRM1 CPU 单元

a）SRM1-C01-V2　b）SRM1-C02-V2

1—外设、RS232 口通信指示灯　2—网络通信状态指示灯　3—CPU 状态指示灯　4—外设口　5—口端盖　6—接线端子

从图知，它的 CPU 单元不集成任何 I/O 点。其面板上也仅有一些指示灯、通信口及接线端子。它的完整配置要靠网络实现，没有网络，也就没有它的控制系统。图 10-3 所示为它的配置实例。

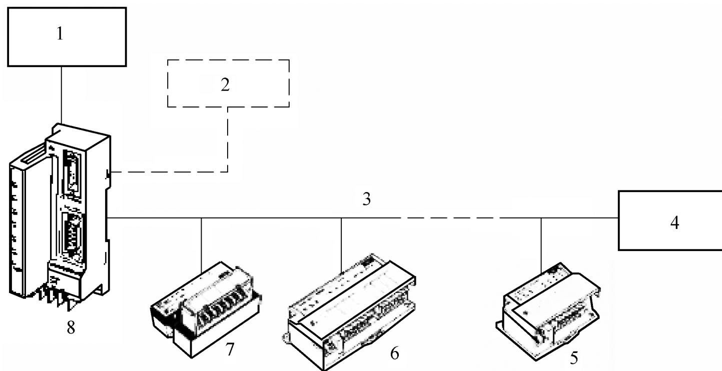


图 10-3　分布式系统基本配置

1—上位计算机　2—外设　3—CONBOBUS/S 通信电缆　4—终端器　5、6、7—从单元　8—SRM1

从图知，支持它的是 OMRON 的部件网（CONBOBUS/S）。所有 I/O 功能都在作为终端的从单元上实现。最多可接入的从单元为 32 个。最多的控制点数为 256 个。

网络所用的电缆是 OMRON 特制的，用其可构成非常可靠及高效的网络系统。数据传输速度很快，系统虽是分布的，但可做到输入响应时间最大也仅需 1ms。

使用 SRM1-C02 CPU，网络范围可达 500m，而且，还可进行模拟量控制。SRM2 比 SRM1 还增加了 RS-232 通信口。

SRM1 CPU 的主要性能如表 10-3 所示。

表 10-3　SRM1 CPU 的主要性能

项　　目	SRM1-C01/C02-V2
指令类型	基本指令　14 条 功能指令　81 条　125 种变化
执行时间	基本指令　0.97μs(LD 指令) 功能指令　9.1μs(MOV 指令)
程序容量	4096 字
最多点数	256 点
定时器/计数器	128(TIM/CNT 000-TIM/CNT 127)
DM 区	读/写:2022 字(DM 0000 ~ DM 2021) 只读:512 字(DM 6144 ~ DM 6655)
内部定时中断	一个,时间间隔(0.5 ~ 319968ms)
备份内存	闪烁内存 无需电池支持,可备份程序及只读 DM 区数据 锂电池支持内存 还可备份读写 DM 区、HR 区、计数器。在 25℃大气温度下 锂电池寿命 10 年
外设口	一个
RS-232 口	一个(SRM1-C02-V2 only)

其网络的主要特性如表 10-4 所示。

表 10-4　网络的主要特性

项　目		特　性
通信协议		CompoBus 协议
传送方法		Multi-drop,T-hrsnch
波特率	高速通信模式	750kbit/s
	长距离通信模式	93.75kbit/s
调制方法		Bascband 方法
编码方法		Manchester 编码方法
最多可连接节点		32:16IN 和 16OUT
		16:8IN 和 8OUT
每帧容量		256(128IN and 128OUT),当连接 32 节点
		128(64IN and 64OUT),当连接 16 节点
通信时间	高速通信	0.8ms,当连接 32 节点
		0.5ms,当连接 16 节点
	长距离通信	6.0ms,当连接 32 节点
		4.0ms,当连接 16 节点
通信距离	高速通信	干线长:100m 最大 单个支线长:3m 最大 支线总长:50m 最大
	长距离通信	干线长:500m 最大 单个支线长:6m 最大 支线总长:120m 最大
电　缆	Vinyl-clad VCTF JIS3306	2 条 0.75mm^2(信号线)
	扁平电缆	4 条 0.75mm^2(2 条信号线和 2 条电源线)

有关终端单元，用的是 OMRON 的产品。其型号及其特性，可参阅有关说明，在此略。

分布 PLC 很适用于像电梯这样，I/O 点分散在各处的系统。使用它可简化系统硬件接线，也便于系统调试与维修。

日本横河电动机也有基于网络的控制的系统 Stardom。它组合了基金会现场总线 H1、安全的互联网、通信技术，以及监控和数据采集（SCADA）软件。Stardom 的性能介于分布式控制系统（DCS）和 PLC 之间，虽然它并不是设计用来代替该公司现有的 DCS。

10.4.4　内装式 PLC

内装 PLC 没有自身外壳包装。在数控系统中，不少开关量控制所用的就是内装的 PLC。只是，它不是 PLC 厂家生产的。而是数控厂家生产的数控系统的一部分。也有的数控系统厂家，如沈阳某厂，为了简化数控系统，过去曾用 OMRON C20P 实现数控 6 角刀架的转位控制。而有了内装 PLC，就比 C20P 方便了。

图 10-4 所示为 OMRON CPM2B 型 PLC 及其与计算机链接通信的情况。这里，CPM2B 可装到其它系统中，但有外设口，通过接口适配器（CPM1-CIF01RS-232C），可与计算机连接。此外，它还有 RS-232C 口，可直接与计算机连接。连接后，可把在计算机上编的程序下载给它。也可直接用外设口接简易编程器，在编程器上编程。

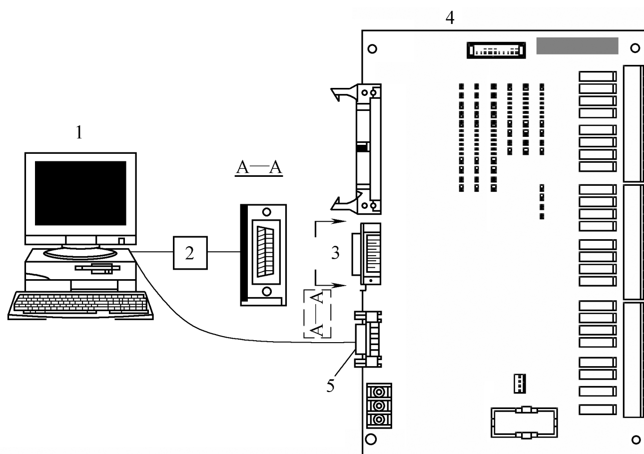


图 10-4　计算机与 CPM2B 链接

1—计算机　2—接口适配器　3—外设口　4—CPM2B（可内装板）　5—RS-232 口

CPM2B 有 60 个 I/O 点，但它有扩展单元，此单元也是可内装的。扩展 I/O 点数为 20。一个主机最多可扩展 3 个扩展单元。这样一个 CPM2B 系统最多可控制的点数为 120 点。图 10-5 所示为 CPM2B 接扩展单元情况。

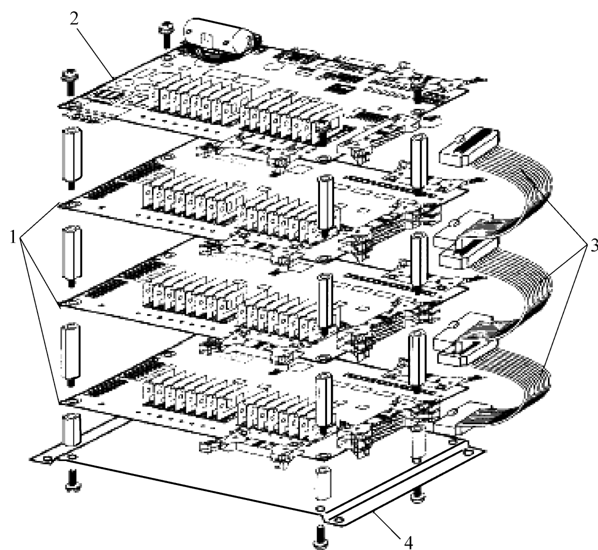


图 10-5　CPM2B 与其扩展单元连接

1—扩展单元　2—CPU 单元　3—连接电缆

4—CPM2B-ATT01 安装架

10.4.5　安全型 PLC

安全型 PLC 是指，自身的关键部分作了冗余配置的，合乎最严格的安全标准的 PLC。美国 AB 公司最早推出这种 PLC。目前推出的型号是 GuardPLC 1200 和 2000 两个系列。它们与现有的 PLC 相兼容。可用它监控和调节一个全自动化系统的安全运行。

它的 CPU 是冗余的。I/O 模块也是特殊的，设有内部测试电路，可自动去检查 I/O 是否正常。该产品通过 TüV 产品周期服务鉴定，符合 EC 61508 标准，支持安全等级达到三级的应用。

它有 3 个基本特点：

1. 结构

结构与普通的 PLC 的不同，有冗余的微处理器。并且，其看门狗电路（Watchdog Circuit）及同步检测电路（Synchronous Detection Circuit）总是不断地对 FLASH 与 RAM 内存进行监控。

2. 输入

输入与普通的 PLC 的不同，其每一输入点都有与其对应的内部“输出”电路，当 PLC 运行时，用周期很短、高低交变的电平，去驱动（Driven）输入点，以检测这些输入点的功能是否正常。

3. 输出

输出与普通的 PLC 的不同，其输出模块有内装的逻辑测试电路，有自己的微处理器。一旦输出点出现故障，PLC 即可得知这些情况，并能有序地进行处理。

可知，这种安全 PLC 把冗余配置，原来处于 PLC 外部，而今转移到 PLC 内部。既减小了体积，成本，简化了系统配置、使用，又提高了工作可靠性。可满足最高等级的安全要求。

除了 AB 公司，Siemens 能源与自动化运动公司也积极参与这一领域的开发，近期也推出它的 SIMATIC S7-400F，一个高端 S7-400 PLC 的全安全版，它可在 SIL 3 安全要求下运行，通过 IEC 61508 认证。在一个临界事件中，该控制器进入一个用户定义的安全状态为一个有序的停止，同时提供诊断数据。系统在 SIMATIC STEP 7 普通开发环境完成编程，使用一个带有与安全相关的功能程序块库。

10.4.6　运动控制用 PLC

运动控制用 PLC 是指专门用于运动控制的 PLC。也称之为可编程运动控制器（Programmer Motion Controller，PMC）。

PMC 与装有运动单元的 PLC 的区别在，前者控制的规模更大。再就是，编程软件前者要完善的多，比较好操作。另外，界面、配套的模块等等也都较齐备。它的目标是与 NC 抗衡的，所以，它总是使自身具有尽可能高的与 NC 竞争的能力。

如日本三菱 Q 系列运动控制器，已是其的第二代的产品。是高性能的运动控制器。运行周期降至 0.88ms，提升了凸轮速度并缩短了操作响应时间。利用三菱 Q 系列 PLC 的多 CPU 功能，Q 系列运动控制器，可以和 PLC 的 CPU 安装在同一基板上，各自发挥它们的优势，将顺序控制和运动控制分开处理，优化系统的结构。

采用多 CPU 组态，最多可采用 4 个 CPU（包括 PLC CPU），一个系统最多可以控制 96 轴。Q 系列运动控制器采用高速串行通信简单设置伺服电动机同步或绝对系统。

同时，运动 CPU 能够通过 SSCNET 与伺服放大器连接，单个运动控制器可以控制伺服放大器最多 32 轴，伺服电动机容量从 10W 到 55kW。

德国西门子也有类似产品。它的 Simotion 将运动控制、逻辑和专门的控制技术融合在一组产品内。

Simotion 有 3 个基本部分。它的两个软件部分是一个称为 Scout 的工程系统，Scout 包括系统配置工具、项目管理工具、程序编辑器以及一个运行时间系统。Simotion 的第三个部分由多个硬件平台组成，目前提供基于控制器的版本（Simotion C）和基于 PC 机的版本

（Simotion P）。即将推出的是用于分散控制的、基于驱动的平台。

据西门子报道，Simotion 通过将各种机器功能集成到 PC、控制器或驱动平台内，因而显著地减小了机器控制器的开发时间和成本。Scout 工程软件能配置所有的平台型号。

OMRON 公司的 FQM1 也是类似产品。

10.4.7 过程控制用 PLC

可编程过程控制器（Programmable Process Controller，PPC）是指主要用于过程控制。PPC 多是在大型 PLC 的基础上，经强化形后成的 PLC。在硬件上，它引入类似 DCS 的操作系统，增强 CPU 的控制及数据处理功能，强化网络接口，并利用 PLC 原有的各种模拟量处理模块；在软件上，它有一系列适合过程控制用的组态、操作界面设计平台，为建立工程师站、操作员站提供方便。

因为用的基本上 PLC 的模块，所以，价格不太高，但也具有 DCS 的功能与性能。

西门子的 SIMATIC PCS7，还有 GE 公司、AB 公司等都有类似产品。

PCS7 就是西门子公司的这样产品。它运用计算机软、硬件技术，在西门子公司 S5，S7 系列可编程控制器及 TELEPERM 系列集散系统的基础上，开发而成的。

PCS7 有以下几个方面的特点，即：客户/服务器的结构；集中的，从上到下的组态方式；集中的，友好的人机界面；含有配方功能的批量处理包；开放的结构，可以同管理级进行通信；同现场总线技术融为一体。

10.4.8 软件 PLC

软件 PLC（SoftPLC，也称为软逻辑 SoftLogic）是一种基于基于 IPC（工业计算机）结构的控制系统。它综合了计算机和 PLC 的开关量控制、模拟量控制、数学运算、数值处理、网络通信、PID 调节等功能。既有硬 PLC 的功能、可靠性、速度、故障查找等方面的优点，又有 IPC 的各种优点。

用它时，用户可以自由选择 PLC 硬件，而不受某个硬 PLC 制造商本身专利技术的限制；可方便地与计算机网络相连；还用自己熟悉的编程语言编程。

已投入市场的软件 PLC 产品较多。据了解，在美国底特律汽车城，大多数汽车装配自动生产线、热处理工艺生产线等都已由传统 PLC 控制改为软件 PLC 控制。

国内有不少公司从事此类产品的开发。有的不用 IPC，而用嵌入式计算机。称嵌入式软件 PLC（Embedded SoftPLC）。

如有的嵌入式软件 PLC，在实现 PLC 功能的同时，还有嵌入式 Web 服务器、FTP 服务器的功能。它的嵌入式网关功能模块和目前在国内广泛使用的 CAN 现场总线实现了无缝集成，从而可以和 CAN 现场总线组成基于 PLC 的分布式控制网络。

使用这样的嵌入式软件 PLC，用户不用去控制现场，在控制室就可以编 PLC 程序，然后通过网络把 PLC 程序下载，并可以实现 PLC 指令的在线更新。

为了保证用户编写 PLC 指令的正确性，用户在把 PLC 指令下载到服务器之前，可以利离线模拟器进行模拟、调试。

国外不少生产普通 PLC 的厂商，也生产此 PLC。如 AB 公司的 SoftLogix 5 Controller 等。SoftLogix 5 Controller 运行在 Microsoft Windows NT 或 Windows 2000 的平台上，提供了 PLC5

的功能及开放式的解决方案。

10.5　PLC 面临的挑战

关键词：集散控制（DCS）、现场总线（FCS）、计算机控制、单片机、嵌入式计算机、数字控制（NC）

在工业控制中，除了 PLC，还有不少其它工业控制。最常用的有集散控制（DCS）、现场总线（FCS）、计算机控制（PCC）、数字控制（NC），等等。这些控制，各出现在不同的年代，各有其特点，各有其运用范围。但都是基于计算机或微处理器、信息处理、数据存储、图像显示及联网通信技术，都是通过运行程序实现控制及信息处理。

这些功能也很强，性能也很高，发展也很快的自动化手段与产品，始终是 PLC 的竞争对手。使 PLC 不断地面临挑战。

它们之间一直在互相吸取对方的优点，扩大自身的运用范围。以至于都说，自己可以取代对方。

事实上，这些控制各有千秋，要说谁一定就能取代谁，在短期内都不大可能。

但要指出的是，PLC 的生命力是强大的。它的强大在于：它控制的范围可大可小，几乎所有的控制领域都可用它；它控制的对象可以是开关量，也可以是模拟量和脉冲量，几乎什么量的控制都可用它；它可用作控制，也可用作数据终端和系统诊断，几乎什么工程任务都可用它，而且，还在于它有世界级的大厂商做支柱，产品可大量生产。所以，产品的价格低。这些大厂家的丰富经验及充足的人才资源、技术资源、财力资源，物力资源，使其有极强新产品开发能力。几乎，年年都有新模块出台，隔几年就有新品种问世。而且，PLC 自身的概念还在不断更新。因而谁想取代它，恐不大可能。但 PLC 真的，实实在在的面临挑战。我们下面介绍几个对 PLC 形成挑战的有关技术。

10.5.1　集散控制系统

集（也有译为分）散控制（Distribute Control System，DCS）主要用于过程控制。它的诞生以 1975 年 Honeywell 公司推出的 TDC2000 系统为标志。关于过程控制，于上个世纪六七十年代占主导地位的是模拟仪表。其显著缺点是：模拟信号精度低，易受干扰。

20 世纪 70 ~ 80 年代占主导地位的变为集中式数字控制系统。用的是数字信号，克服了模拟仪表精度低的缺陷，并提高了系统的抗干扰能力。

集中式数字控制的另一优点是，易于根据全局情况进行控制计算和判断，在控制方式、控制机时的选择上可以统一调度和安排。不足的是，当系统任务增加时，控制器的效率和可靠性将急剧下降。

于 20 世纪 70 年代中期出现了集散控制（DCS）。它的核心思想是，集中管理、分散控制。上位机集中进行监视、管理，分散到现场的下位机，实施控制。上、下位机之间用网络互连，进行信息交换。因此，这种结构减少了出现故障的风险。

同时，DCS 功能多样、操作简便、系统便于扩展、维护方便、可靠性高、便于与其它计算机联用等特点，所以尽管其价格比其它控制产品要贵一些，系统的开放性远比其它产品差，但由于其可靠性和安全性及性能等方面的优点，还是获得了广泛的应用。

DCS 自问世以来，大约有三次比较大的变革。20 世纪 70 年代，它的操作员站的硬件、操作系统、监视软件都是专用的，由各 DCS 厂家自己开发，并且设有动态流程图。通信网络基本上是轮询方式的；80 年代，它的通信网络较多使用令牌方式；90 年代，它的操作员站出现了通用系统，90 年代末，它的通信网络有的部分遵循 TCP/IP 协议，有的开始采用以太网。20 多年以来，DCS 已经广泛应用于各工业领域并趋于成熟。

发展到今天，DCS 已将开放性作为一个必要的指标，虽然各生产厂商的 DCS 不能直接相连，但各 DCS

由于采用了国际标准化的网络和通用的数据库以及其它技术，因此各 DCS 的相连不再是件难事。另外，早期的 DCS 一般尽量采用成熟的计算机技术，而以技术保守著称，而今的 DCS 厂商为提高系统的计算性能和竞争实力争相采用最先进的计算机技术。过去的 DCS 最多的是为用户提供一个控制系统平台，用户通过组态实现过程控制功能，而当今的 DCS 几乎都增加了综合管理功能，以实现全厂综合自动化为目标。

出自石化、冶金、电力等行业的改造和新建工程的急需，于 20 世纪 80 年代，我国从国外引进了几百套 DCS，其型号大约不少于 60 余种。这些系统投资高，中、小型企业很难承受。为此，进入 90 年代后，我国的一些厂家也已开发或正在开发适合我国中、小型企业使用的小型 DCS。这些产品价格较低，备品、备件齐全，技术也较先进，同时，它的组态学习容易、技术支持也很得力。

DCS 的基础是系统网络。因此，网络的实时性、可靠性、扩充性及重构性一定要好。

系统网络的实时性是指，必须在确定的时间内完成所需信息的传送。衡量这个性能指标，不仅仅是用网络的速率（波特率），而是要看是否能在要求的时间内，确保所需信息的传输。

系统网络可靠性是指，无论在任何情况下，网络通信都不能中断。因此多数 DCS 是采用双总线、环形或双重星形的网络拓扑结构。

系统的扩充性是指，可接入的最大节点数量，应比实际使用的节点数量大若干倍。这样，一方面可以随时增加新的节点，另一方面也可以使系统网络运行，处于较轻的负荷状态，以确保系统的实时性和可靠性。

系统网络重构性是指，在系统实际运行过程中，各个节点可任意地上网、下网，而不影响系统的正常运行。

DCS 通过现场网络节点实施对 I/O 控制或数据采集。一般在一套 DCS 中，要设置多个现场 I/O 控制站，以分担整个系统的 I/O 控制功能与数据采集功能。

此外，DCS 还有操作员站，它是人机界面（Human Machine Interface，HMI 或 Operator Interface）的网络节点。操作员可使用它处理与操作有关的事务。

再者，DCS 还有工程师站，它是对 DCS 进行离线的配置、组态工作和在线的系统监督、控制、维护的网络节点。其功能是对 DCS 进行配置、组态；并在线监视 DCS 网络上各个节点的运行情况，调整系统配置及一些系统设定参数，使 DCS 处在最佳的工作状态下工作。

DCS 的控制及数据处理能力是很强的，网络速度是很高的。由于它可进行完全冗余配置，故可靠性也是很高的。同时，它可设立较多的工程师站、操作员站、服务器站，并可配备相应软件，故系统配置及实际操作也都很方便。这些多是 PLC 所不及的。

但是，在集散控制系统中，分布式控制的实现得益于网络技术的发展和应用。遗憾的是，目前这个通信网络都是各自专用的，封闭的，不同厂家的 DCS 系统之间难以实现网络互连和信息共享。

此外，DCS 还有以下一些不足：

首先是其一对一结构，即一台仪表，一对传输线，单向传输 1 个信号。这种一对一结构造成接线庞杂、工程周期长、安装费用高、维护困难。而模拟信号传输不仅精度低，且易受干扰。为此采用各种措施进行提高抗干扰性和传输精度，其结果是增加了成本。

其次，是失控状态。操作员在控制室既不了解模拟仪表的工作状况，也不能对其进行参数调整，更不能预测事故，导致操作员对其处于失控状态。由于操作员不能及时发现现场仪表的故障，而发生的事故已屡见不鲜。

再次，互操作性差。尽管模拟仪表统一了 4～20mA 的信号标准，可大部分技术参数仍由制造商自定，致使不同品牌的仪表无法互换。因此导致了用户依赖制造厂，无法使用性能价格比最优的配套仪表，甚至出现个别制造商垄断市场的局面。

最后，DCS 的组成系统庞大，价格昂贵。开关量控制功能差，故只适用于系统较大、主要为过程控制、工作可靠要求较高，而投资实力又较雄厚的场合。

随着技术发展，DCS 技术也在发展：正向综合方向发展，正将各种单（多）回路调节器、PLC、工业

PC、NC 等工控设备构成大系统，以满足工厂自动化要求，并适应开放式的大趋势；正向智能化方向发展，正使用以微处理器为基础的各种智能设备，如智能 I/O、智能 PID 控制器、智能传感器、变送器、执行器、智能人机接口、可编程调节器等等；正向工业 PC 化发展，IPC 成为 DCS 的硬件平台；正向专业化发展，正形成如核电 DCS，变电站 DCS、玻璃 DCS、水泥 DCS 等。

PLC 与 DCS 相比，其功能、可靠性、控制能力及使用方便，特别在软件的配套上差距是不小的。但 PLC 开关量控制方便，价格低，可从别的领域，包括 DCS，吸收长处，发展很快。DCS 有的技术，它也逐步有了如网络化、冗余配置、配套软件，现也已越来越多样，也越来越完善了。

10.5.2　现场总线控制

现场总线系统（Fieldbus Control System，FCS）是继 DCS 之后出现的又一种新型的工业控制系统。它改变了原有的控制体系结构，使模拟与数字混合的集散控制（DCS），更新为全数字的控制，用开放的现场总线网络，将现场各控制器及智能仪表互联，构成可互操作的现场总线控制系统。它把控制功能彻底下放到现场，真正做到控制分散、危险分散、集中监控和全数字化。因此，FCS 实质是 一种开放的、可互操作性的、彻底分散的、分布式控制系统，比 DCS 又前进了一步。

按照国际电工委员会 IEC 标准和现场总线基金会 FF 的定义，现场总线是连接智能现场设备和自动化系统的数字式、双向传输、多分支结构的通信网络。由于现场总线适应了工业控制系统向分散化、网络化和智能化的发展趋势，它一经产生便成为全球工业自动化技术的热点，受到了全世界的普遍关注。

图 10-6 所示为一种现场总线结构简图。

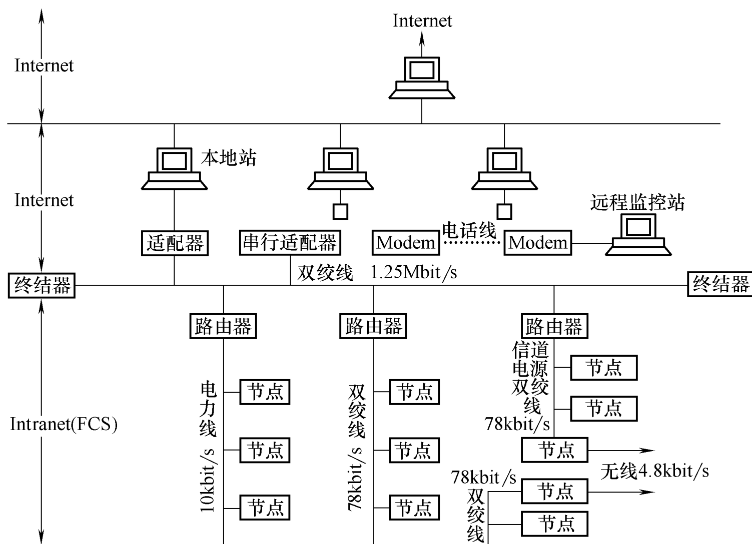


图 10-6　现场总线结构

从图知，它的最底层为 Intranet 控制网，即 FCS。各控制器节点都放到现场，构成一种彻底分布式控制结构。其网络拓扑结构任意，可为总线形、星形、环形等。通信介质也不受限制，可用双绞线、电力线、无线、红外线等。FCS 形成的 Intranet 控制网很容易与 Intranet 企业内部网和 Internet 全球信息网互连，构成一个完整的企业网络三级体系结构。

根据 IEC 标准及现场总线基金会的定义：现场总线是连接智能现场设备和自动化系统的数字式双向传输、多分支结构的通信网络。它的基础是所有仪表、装置的智能化，而且有通信口，能接受相应通信协议。

国际电工协会（IEC）的 SP50 委员会对 FCS 有以下三点要求：

（1）在同一数据链路上，过程控制单元（PCU）、PLC 等与数字 I/O 设备互连；

（2）现场总线控制器可对总线上的多个操作站、传感器及执行机构等进行数据存取；

（3）通信媒体安装费用较低。

FCS 是全数字化的通信，从传感器、变送器到调节器，均为数字信号。这使得复杂、精确的信号处理得以实现，并提高了控制的可靠性。其特点是，系统开放性好、互可操作性与互用性好、设备具有智能化与功能自治性、结构的高度分散性、对环境的强适应性、系统成本低。

FCS 发展迅速。目前已经开发出 40 多种现场总线，如 Interbus、Bitbus、DeviceNet、MODbus、Arcnet、P-Net、FIP、ISP 等。其中最有影响力的 5 种分别为 FF、Profibus、HART、CAN、LonWorks，如表 10-5 所示。

表 10-5　常用现场总线主要特性

特　　性	现　场　总　线　类　型				
	FF	ProfiBus	HART	CAN	LonWorks
应用范围	仪表	PLC	智能变送器	汽车	楼宇自动化、工业自动化等
OSI 网络层次	1,2,3,8	1,2,7	1,2,7	1,2,7	1～7
通信介质	双绞线、电缆、光纤、无线等	双绞线、光纤	电源信号线	双绞线、光纤	双绞线、电力线、电缆、光纤、无线等
介质访问方式	令牌、主从	令牌、主从	令牌、查询	位仲裁	P-P CSMA
纠错方式	CRC	CRC	CRC	CRC	CRC
通信速率(Mbit/s)	2.5	1.2	1.2	1	1.25
最大节点数	32	128	15	110	2^{48}
优先级	有	有	有	有	有
保密性	—	—	—	—	身份认证
本安性	是	是	是	是	是
开发工具	有	有	—	有	有

由于可预见到现场总线的优越性，国家对这个技术的开发是很重视的。如果它能开发越来越多的功能模块，以及能使有关参数标准化，产生公认的、开放的标准，那将大大推进这个技术的推广与应用。这将给控制系统的结构带来革命。有人甚至预言，21 世纪是现场总线的世纪，PLC 将退出历史舞台。

然而，笔者对此不敢苟同。尽管现场总线有其分散性、开放性等种种优点，但它与采用 PLC 并不矛盾。FCS 的出现并不要求 PLC 退出历史舞台。更何况 PLC 也还有很强的生命力，自身仍在不断前进着。只要 PLC 厂商能协调各自的商业利益，在建立公认的标准，加强系统开放性上认真地做些工作，完全可实现自身的开放及与其它控制（含 FCS）的兼容。可成为现场总线中的一个站点，也可把现场总线中的诸站点处于它的管理之下。

更何况现场总线本身还是一种正在发展中的技术，在许多方面还需要改善。IEC61158 规定了 FF、Profibus、WorldFIP 等 8 种现场总线标准，还有一些事实上的标准，如 LonWorks 和 CAN 总线等。现有 8 种现场总线为国际标准，它们采用的通信协议完全不同，因此要实现这些总线的兼容和互操作是十分困难的。许多标准的并存，将导致现场总线技术不易广泛应用。

另外，现场总线还存在瓶颈问题，如：总线切断后，系统有可能产生不可预知的后果。在防爆应用中，现有的防爆规定限制总线的长度和总线上所挂设备的数量，同时也限制了现场总线节省线缆优点的发挥；系统组态参数过于复杂，不易掌握，而其设定的优劣对系统性能的影响很大。

当然，现场总线也正在发展中。如现场总线 FF，2000 年 3 月宣布将采用以太网增强 H2 的方案。因

此，一般认为整合 Ethernet 和 TCP/IP 技术的现场总线是今后发展的主流体系和应用热点。

10.5.3　工业计算机控制

工业个人计算机（Industry Personal Computer，IPC）自 20 世纪 90 年代初进军工业自动化以来，正势不可挡地进入工业控制的各个领域，获得了广泛应用。

IPC 实质上就是个人计算机。只是按工业现场条件与控制使用的要求，在结构与系统构成上作了改进。它既保留了个人计算机功能强、软件丰富、外设多、开放性、发展快的优点；又采取了很多隔离与保护措施，增强了硬件可靠性，并增加了种种 I/O、A/D、D/A、温度、高速计数等插板及通信接口卡，以及很多远程实时控制组件，故也适用于在工业环境下，进行种种控制与数据处理。

它的问题是：性能过剩，性价比低；插板品种、规格少，不便配置；接插麻烦，也难以即插即用；体积较大，给在现场安放也带来不便。

但是，情况也在变化。过去专用的封闭式的架构迅速被 PC 技术的开放式架构所取代，其中最重要的原因之一就是软件的兼容性。工业控制机用户过去和现在都希望使用与所熟悉的桌面 PC 相同的操作系统和开发工具，这就导致了开放式桌面 PC 在工业环境中的直接应用。然而，桌面 PC 技术是面向每年 2000 亿美元的商用机市场的，而不是相对比较小的工业计算机市场的，它不具备工业级性能，如抗强振动、冲击、高温、潮湿和具有快速修复能力（MTTR 一般小于 5 分钟）。由桌面 PC 技术衍生的 ISA 总线加固型计算机在工业上得到了相当广泛的应用。

另外，新一代的工业计算机的结构及系统组成已有了很大变化。结构形式也已多样化，系统组成也很灵活。除了传统的台式，增加了平板式、机箱式、框架式等。有的也是按单元组装。外形颇似大型的 PLC。已一改工业 PC 与个人计算机相差不多的形象。

如 MIC2000，是基于 ISA 总线的无源底板技术代替有源底板。所有单元包括 CPU、硬盘、软驱、I/O、通信等，都用模块化设计，系统可任意组合。机架上的槽也可多可少（目前仅开发有 8 槽、11 槽）。而且，一反过去工业 PC 由后端接线，而改为前端接线。模块是从前面抽拉，这使得模块的安装与取出变得很简单，从而可大大缩短维修时间。且用了可插拔式的端子排，与当今世界上流行的工业控制结构一致，使得接线与维护如同 PLC。

它的所有模块都是四面固定的，能经受恶劣的工业环境的强烈冲击和振动。另外，而且，还备有 5V、12V 的直流电源，供用户使用。

据介绍，在 70% 的负载下，其平均无故障时间可达 50000 小时。可见，IPC 的发展也吸取了不少 PLC 的特点或优点，使自身能更好地适应工业环境，更便于使用。

尽管如此，在系统不大时，IPC 怎么也是难以与 PLC 相比的。功能过剩，价格较高总是难以避免的。为此，近年来还推出紧凑型 PCI 总线工控机（Compact PCI）。

Compact PCI 是一种新的开放式工业计算机标准，它是 PCI 总线技术和成熟的欧式卡组装技术的结合。采用 CompactPCI 既能吸收 PC 机最新的技术成果，又具有满足通信和工业实时应用所必需的更坚固、更可靠、模块化、易使用、易维护的优点。

CompactPCI 最具吸引力的特点是热插拔（hot swap）。这意味着一个模板能够在不切断电源的情况下从机箱内拔出或插入，依靠先进的软件，系统能够自动调整配置。热插拔一直是电信应用的要求，也为每一个工业自动化系统所渴求。

……

IPC 可以自成控制系统。同时，它组成的系统，在管（车间、工厂管理）控（设备、生产线控制）一体化上还有其独特优势。管控一体化将更加有效地提高企业生产率，增强企业的市场竞争能力。

IPC 也可成为 DCS、FCS 中的工程师站、操纵员站的基本设备。还可成为 PLC 控制系统的上位监控站。IPC 的这方面应用是很可观的。对 PLC 讲，除了可编程终端（当然，如允许用 PC，则 PC 更合算）有类似 IPC 的功能外，也是别的控制设备所无法替代的。

再就是近期出现的软件 PLC，实质上就是 IPC 控制系统。只是引进了 PLC 的理念，配置了 PLC 一些 I/O 或远程模块，使用 PLC 的编程语言，如梯形图语言，按照 PLC 编程方法编程。软件 PLC 既可算是新型 PLC，因为它确实具有 PLC 的种种特点；又可算是新型 IPC，以为是实质上它确是 IPC。

正是 IPC 的这些多方面应用，所以，近年来，IPC 市场增长强劲。每年以 20% 以上的速率增长，远远超出了 DCS、PLC。值得注意的是，各大 PLC 制造商已接受了 IPC 的技术路线，PLC 有 IPC 化的趋势。不仅如此，DCS、NC 也都有 IPC 化的趋势。PC-based 控制技术也许可能将成为本世纪初自动化界的主流技术之一。

IPC 蓬勃发展的还表现为：各大 PLC 制造厂商，如 Siemens、Rockwell Automation、GEFanuc、三菱电机均推出自己品牌的 IPC 产品。

西门子公司于 2002 年 3 月推出 SIMATIC“速溶咖啡”，是基于 PC 的自动化套件。它以西门子工业 PC——机架式 PC 或面板式 PC 作为硬件平台，整合基于 PC 的控制器 WinAC，人机界面软件 WinCC 或 ProTool/Pro，编程组态工具（STEP 7）和 Profibus DP 通信卡 CP5611/5613 等产品。用户可直接将套件通过 Profibus DP 电缆连接分布式 I/O 模块 ET 200 构成 PC 平台上全集成的自动化解决方案。图 10-7 所示即为它可能一个配置的方案。

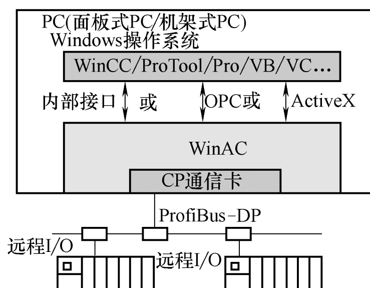


图 10-7 可能的配置方案

从图 10-7 可知，它有相应的控制算法、数据采集与信息集成软件，可进行通信和 I/O 处理、运动控制、视频控制。适合于在汽车工业、食品饮料、烟草、市政环保、电力、水处理、物流仓储、半导体生产、机器人、智能楼宇、仿真教学等行业中使用。

当然，既然 IPC 可吸收 PLC 的优点或特点，PLC 为什么不反过来也这么做，也吸收 IPC 的优点，进一步使自己完善起来呢？事实上，近年 PLC 在扩展存储能力，以至于也发展计算机单元、ASCII 单元、硬盘单元、软驱单元等等，也正是这种趋势的体现。

10.5.4 其它控制

其它用于控制的技术还有单片机、嵌入式 PC、NC。也是向 PLC 挑战的对象。

1. 单片机

单片机指集成有 CPU、I/O 接口及内存的芯片。尽管仅是一个芯片，但基本上是一台计算机。单片机有 4 位的，8 位的，16 位的。大的也有 32 位的。单片机功能很强，位多的功能更强。使用单片机，加上必要 I/O 设备及通信接口，就可以利用信息处理的办法，实施对系统控制、数据采集及联网通信。

单片机价格低廉，功能强大，获得广泛的应用。单片机系统可有针对性地开发，很适用于较大批量生产的产品。故在智能家电产品中，单片机用的很多。一台洗衣机，配置了单片机控制系统，就可按程序工作，实现自动化、智能化。工业设备配备单片机实现自动化的例子更多。

PLC 实施控制，其基本原理也与单片机相同。有的 PLC 的 CPU，如东芝小型 PLC EX40，其 CPU 芯片用的就是单片机 8031（不带内存），自身再配上内存，并开发相应的监控程序（即操作系统），就成了它的 PLC 了。只是，在它成为产品之前，要作很多测试，同时，还作了很多加强安全的可靠性设计。

为此，有的电梯厂原来用 PLC 替代继电器控制。而如今还有想用单片机替代 PLC 进行控制。用单片机控制比 PLC 价格要低得多。还有各住宅小区的恒压供水控制，不少使用 PLC，但也不少用单片机。关键还在单片机价格低，使用起来也较灵活。

但是，单片机的可靠性还是差些，系统的构建也比较麻烦。不如使用 PLC 可靠、方便。特别是对最终用户而言，这正如，自己装收音机当然也能用，但怎么说，质量是不如专业厂家生产的收音机，而且，也很费事。除了出自学习的目的，谁还会那么做？所以，不是大批量的应用，很少再用单片机了。

用单片机的另一问题是维护困难。很多电梯厂用单片机控制，由于人员变化，售后服务就很吃力。有的已深感还是用 PLC 方便。

当然，单片机也在进步，特别在编程上，有的现也可使用高级语言，单片机用的 C 语言，编程。这比原先用汇编语言要好的多。

在硬件上，单片机也在进步。现也能够将功能复杂的众多外围功能部件，全部或大部分集成到系统所使用的单片机内部，以提高工作可靠性。同时又可使系统的成本得以降低。美国 Cygnal 公司的 C8051F020 单片机便是这样的单片机。

C8051F020 单片机具有与 MCS 51 内核及指令完全兼容的微控制器。除了具有标准 8051 机的数字外设部件外，片内还集成了数据采集与控制系统中常用的模拟部件和其它数字外设及功能部件。具体包括模拟多路选择器、可编程增益放大器、ADC、DAC、电压比较器、电压基准、温度传感器、SMBus/I2C、UART、SPI、可编程计数器/定时器阵列、定时器、I/O 端口、电源监视器、看门狗定时器和时钟振荡器等，且该单片机内部具有 JTAG 和调试电路，通过 JTAG 接口可以使用安装在最终应用系统产品上的单片机进行非侵入、全速及在系统调试。

单片机这些高度集成的进步，在 PLC 的系统也是有所体现的。如今 PLC CPU 单元的核心芯片也是高度集成的。所以，单片机的进步，既是对 PLC 的挑战，也给 PLC 提供了技术提升的借鉴。

2. 嵌入式 PC

嵌入式 PC 比起单片机，则是更前进了一步，嵌入式系统是集软件、硬件于一体的高可靠性的小型计算机系统。它可“嵌入”（Enbeded）到设备内部，提供用户接口，管理数据输入、输出、控制设备工作的计算机。它的软件、硬件都可灵活裁剪，以适应应用系统对功能、可靠性、成本、体积、功耗的要求。其软件除操作系统外，还需有固化的应用软件；硬件除了 CPU 外，还需有外围电路支持。可以这么定义嵌入系统，即：嵌入式 CPU + 嵌入式 OS + 关键应用 = 嵌入式系统的核心技术。

目前，在嵌入式设计中使用最多的还是 80386 和 80486 等，这种 CPU 性能价格比高，有 In-Tel、IBM、Cyixs、AMD 和 TI 等著名 CPU 制造商支持。丰富的软件（包括操作系统、开发工具和应用软件）支撑。

嵌入式系统技术日益完善，32 位微处理器，即 X86CPU 和嵌入式 CPU（ARM、MIPS、SH 三大类）以及 DSP、单片系统 SOC，在嵌入式系统中占主导地位；嵌入式操作系统已经从简单走向成熟（如微软的 WindowsCE、VxWorks、PSOS、VRTX、Nucleus、WinCE、QNX、Linux），嵌入式系统与网络、Internet 结合日益密切。其中，嵌入式 Linux 在消费电子设备中得到广泛应用。由 32 位嵌入式 CPU 与嵌入式 Linux 相结合，在信息终端、家用电器、工业控制等方面将大有可为。

嵌入式系统的出现至今已经有 30 多年的历史，近几年来，计算机、通信、消费电子的一体化趋势日益明显，嵌入式技术已成为一个研究热点。嵌入式系统已发展到以 Internet 为标志的嵌入式系统。这是一个正在迅速发展的阶段。目前大多数嵌入式系统还孤立于 Internet 之外，但随着 Internet 的发展以及 Internet 技术与信息家电、工业控制技术结合日益密切，嵌入式设备与 Internet 的结合将代表嵌入式系统的未来。

PC 机主要应用在办公室自动化领域，而嵌入式系统已经广泛地渗透到人们的工作、生活之中，从家用电器、手持通信设备、信息终端、仪器仪表、汽车、航天航空、军事装备、制造工业、过程控制等等，无不是嵌入式系统的应用领域。根据美国嵌入式系统专业杂志 RTC 报道，21 世纪初的 10 年中，全球嵌入式系统市场需求量具有比 PC 市场大 10 至 100 倍的商机。嵌入式系统的应用非常广泛，可以说，嵌入式系统无所不在、无时不在，可应用于人类生活与工作的各个领域。未来学家尼葛洛庞蒂曾预言，嵌入式智能设备将是 PC 和因特网之后最伟大的发明。

嵌入式 PC 能在恶劣环境下（如高温、潮湿和震动等）长期可靠工作。嵌入式 PC 平均无故障时间（MBTE）为 100000 ~ 150000h，而台式机仅为 10000 ~ 15000h。硬件、软件和工业 PC 差不多。

当今，市场上每年销售大约 30 亿颗嵌入式 CPU，其中绝大部分是 8 位和 16 位的 CPU。与此同时，嵌入式工业控制机的市场也在迅速增长。嵌入式工控机是嵌入在工业系统内部，在工业极端环境里能够连续长期稳定可靠工作的工业型计算机。

我国在开发嵌入式 CPU 有：

北京中芯系统公司的“方舟”二号，主要技术性能是：32 位 RISC CPU，时钟频率 266MHz，2x8kB cache，片上集成了 EMI 存储器控制器，PCI 总线控制器，中断处理器，DMA 控制器，2 个串行接口，2 个 USB 接口，1 个 IC 卡控制器，1 个以太网控制器，RTC，定时器，Watchdog 等，低功耗，是高集成度的 SOC 芯片，支持 Linux。

中科院计算所的“龙芯”，也是 32 位，266M 钟频，其指令系统与 MIPS 芯片兼容，支持 VxWorks、Linux 操作系统，工作温度为 -35~70℃，可以满足军工和工业环境的应用要求。

“龙芯”和“方舟”的 32 位嵌入式 CPU 的研制成功，标志着我国在嵌入式 CPU 方面取得了很大进展。它再配以嵌入式 Linux 操作系统，将有可能作为国产嵌入式系统的一种很好的选择，可在信息终端、家用电器、工业控制、军工装备中应用。

1999 年 1 月 11 日，世界著名的技术市场研究与战略公司 VDC（Venture Development Corporation）公布的实时/嵌入式模板市场对 OEM 和最终用户需求（Real-Time/Embedded Board Market Needs for OEMs and End Users）报告指出：“实时和嵌入式应用市场 1998 年为 25 亿美元，预计 2000 年为 30 亿，2002 年将达到 39 亿美元”。“今天，安装量排在前几位的工控机是 VME、专用机（无标准总线）、AT96（ISA/PCI）、PC/104、STD/STD32、RISC 模板（无总线）、CompactPCI”。预计“到 2002 年，总线型嵌入式工控机在市场上的占有率分别为：VME 42.6%、AT96（ISA/PCI）36.6%、CompactPCI 4.7%、PC/104 3.8%、STD 0.4%”。在 1998 年，“世界上最有影响的嵌入式工控机 OEM 厂商排名前 6 名的是：Motorola、Ampro、Force、Ziatech 和 Winsystems 公司”。

嵌入式工控机技术不断在向高性能、高可靠、低价位方向发展，将在通信、工业控制和工业自动化、军事和国防、医疗器械、交通运输和模拟仿真等嵌入式应用领域发挥更重要的作用。

嵌入式的发展也给 PLC 带来新的挑战。如本书第 6 章介绍的 CQM1 用于电量采集，用了多年，但终因嵌入式系统的进步，现已被其取代。原因是，嵌入式系统数据存储量大，其它性能也不错，最重要的是它可配置以太网卡，可挂接到互联网上。用普通的浏览器即可快速、方便地读取或上载它采集的数据，而不用拨号、通过电话线速度不高的通信。这当然也给 PLC 的技术提升带来借鉴。

3. 数字控制（NC）

NC 技术是用数字信息对机械运动和工作过程进行控制的技术，是上世纪 50 年代发展起来的。计算机问世不久，它就已开始开发，1953 年就用于机床的控制，既提高了机床加工型面的精度、提高了生产效率，又简化了的模型制造等生产准备工作。NC 是发展新兴高新技术产业和尖端工业（如信息技术及其产业、生物技术及其产业、航空、航天等国防工业产业）的最基本的装备。是当今先进制造和装备的最核心技术。

NC 是综合性的技术，难度大，范围广。牵涉到的有：机械制造技术；信息处理、加工、传输技术；自动控制技术；伺服驱动技术；传感器技术；软件技术等。

几十年来，NC 技术在诸多方面有了很大进步，正向“高速、高精、高效、复合和网络化”方向发展。其技术的主流是开放和智能。新近发展的 STEP-NC，提出一种新的制造理念，在新标准下，NC 程序可以分散在互联网上，使数控技术向开放式、网络化发展。同时，STEP-NC 数控系统还可大大减少加工图纸（约 75%）、加工程序编制时间（约 35%）和加工时间（约 50%）。

我国数控技术起步也较早，于 1958 年，就开始自主研究。但发展较为缓慢。“六五”、“七五”期间以及“八五”的前期，从国外引进技术，消化吸收，初步建立起国产化体系。在“八五”的后期和“九五”期间，实施产业化的研究，我国国产数控装备的产业化取得了实质性进步。到了“九五”末期，国产数控机床的国内市场占有率达 50%，配国产数控系统（普及型）也达到了 10%。

但 NC 造价昂贵，特别是不太复杂的工作系统，用它是不上算的。

NC 技术中，特别是 NC 机床，也用到 PLC 技术，只是它多为内装的；但目前也有用外装的，即用市场上可购到的 PLC。其实，对用户而言，通用的 PLC 维修更方便。

PLC 也开发有 NC、MC 单元，而且种类越来越多，以至于专门用于运动控制的 PMC。但价格要低得多。如果仅用点位控制，如立体仓库的货位控制，或汽车停车场的车位控制，用 PLC 下的 NC 模块是很经济的。但也要看到，越来越多普及型的 NC 出现，必将对 PLC 的在运动控制领域的应用带来新的挑战。

自动控制技术中还有智能仪表、变频、驱动控制、伺服系统等，发展也很快。但这些都是基础一级的，最多只是现场总线下的一个站点，或 PLC 网中的一个站点。它们越发展，越为 PLC 的应用奠定基础，而不会与 PLC 的发展、应用相排斥。

10.6　PLC 去向何处

关键词：进一步完善、进一步分散、进一步开放、进一步便利、进一步热备、进一步趋同、PAC

达尔文在《进化论》里提到，那些能够生存下来的物种，并不是那些最强壮的，也不是那些最聪明的，而是那些对变化做出快速反应的。PLC 的产生、发展甚或消亡也不例外。面对自动控制世界里的 DCS、现场总线、工业 PC、单片机…多种技术，PLC 能生存下来吗？会被取代吗？很多人问这个问题。

我想，被取代的地方与领域肯定是有的。但不会在所有的地方与领域都被取代。或者说，如果 PLC 自身不再发展、不再前进，有可能是那样的。但事实不是那样，PLC 也正在发展，也正在前进。所以，不会在所有的地方与领域都被取代。相反，它还会在它合适的老地方与领域坚守阵地；也会在它合适的新地方与领域去取代别人，开辟新阵地。

究其原因，没有哪一家技术的支持商愿意自动退出历史的舞台，而是不断的吸收新的技术提升自己，以便继续在市场上生存甚至壮大。现在谈 PLC 灭亡似乎为时太早。连已服役了一百多年的，PLC 问世就想要取代它的继电器，现在也还在进步，至今也还没灭亡。何况才有 30 多年发展历史的 PLC 呢？

这正如新的交通工具出现与发展，并不意味着旧的一定就要灭亡。飞机很好，能完全取代火车吗？汽车好，能完全取代自行车吗？有了自行车，难道就不要步行了吗？这叫各有各的用场，各得其所。同理，PLC 与其它控制技术也是各有各的用场，也将各得其所。

当然，由于商业竞争的原因，PLC 的市场不一定始终是处于那么高的速度在增长，甚至也可能有所萎缩。但这不等于灭亡。争论灭亡不灭亡，有个看问题的角度问题。有人从某个行业看，如电厂，它生产的连续性、可靠性，协调性要求很高，控制规模很大。既要求统一协调的集中管理，又要求风险分散的可靠控制。这样场合，可能用 DCS 或现场总线会更好些。PLC 可能将退出这个领域。反之，有的领域，如装备制造业，用 PLC 可能更合适些。所以，PLC 在某个领域被取代，并不意味着 PLC 灭亡。反之，有的领域没被取代，或甚至有新的领域被 PLC 取代，也不意味着 PLC 没有面临危机的可能。大家还是都不要以偏概全的好，最好先别急于下这样“盖棺定论”的结论。

但不管怎么说，PLC 去向何处？还是要讨论的。是否有如下几点值得注意呢？

1. 进一步完善

PLC 应从基础技术与相关技术中，吸取新成果，进一步使自身完善，像本章第 1 节指出那样，继续前进。使自身的功能更强大，使用更方便，工作更可靠，性能价格比进一步提高。这些虽是量变，但量变是质变的基础。有了一点一滴小进步的积累，PLC 定将有更大的

飞跃。

2. 进一步分散

除了自身进一步完善，PLC 控制系统也将进一步分散化，即走向分散控制。这也是 PLC 吸收 DCS 的思想用于自身发展的趋势。

分散控制的优点是既可分散控制风险，又可简化硬件接线，便于维护与修理。如电梯控制，若用 CQM1 的远程系统，或用 C200Hα 机的 SRM 主控，可以做到各层按一个小终端，而终端与主机仅两根通信线相连，可大大简化接线。

如果采用 GE 公司的 GENIUS I/O 模块，或 OMRON 的带 CPU 的 I/O 模块，对就地的I/O 可自行变换，必要时，才与上级站点或相邻站点通信，互操作或交换数据。建立一些这样可“自治”的、智能化的、分散化的站点，然后再用联网的方式进行管理，则也可做到控制“彻底”分散，避免集中管理的危险。

分散化与网络的进步是密切相连的。如果 PLC 具有完善地实现其基本功能的同时，还具有这样的网络技术，即都能配备有类似以太网那样的接口，都能直接，或通过网关灵活地与相应网络相接，都能快速地实现与通信对象交换数据、相互操作。那不就是既降低风险，实施分散控制，而又畅通信息，便于集中管理了吗。

3. 进一步开放

分散化的更进一步发展，将是开放化。因为分散的站点是将是五花八门的。而这些站点的建立，则要靠开发有种种智能化的、自治的、开放的可处理开关量、模拟量及数字量的，有通信能力的小型化单元或装置。显然，这些单元或装置的开发，靠一家公司是难以包打天下的。而没有众多的这种产品，分散化也难以实现。

PLC 厂商出自商业利益，原来必须封闭发展，以后也许要走向它的反面，即走向开放。事实上，这个趋势已经显现。如 GE FANAC 的 90-70 型 PLC，若有合乎 VME（欧洲的工业标准）的一些第三方模块，也可组合进它的系统。OMRON CompoBUS/D 网，只要合乎 DEVICENET 标准的一些智能仪表或装置，也可入网。OMRON 近期还生产有与 ProfiBus 相联接的模块，以期把它的产品也能与西门子的产品相接。

美国有个公司叫 PROSOFT。专门生产这样的模块，一边，可与 AB 或 MODICON 的产品衔接，另一边，则可与西门子的 PLC 衔接。利用它即可把这两不同的产品配置在一个系统中。

所有这些都说明，开放化，允许别人的接进来，或争取接到别的系统中去，已是趋势。

当然，目前这些开放还很不够。有的公司甚至连通信协议都不公开，要用它的软件才能建立自己的应用程序。显然，这是很不适应时宜的。

但愿 PLC 能像个人计算机那样，系统是开放的。那样，不仅用户可更自由地选用它，而且也可减少 PLC 的开发费用，或避免重复开发，从而更有利于 PLC 自身的发展。

4. 进一步便利

PLC 的使用是很方便的，特别是它的系统的建立，比 IPC、DCS、NC 都方便。但它的人机界面不够丰富。IPC 可使用多媒体，PLC 就不那么容易。用它建立的系统，多使用操作台操作。状态及数据显示多还是用指示灯，手动操作多使用按钮等等。总有些不便。

PLC 的这个缺陷正在克服之中。它通过 ASCII 单元，以至于通过计算机单元，可间接实现多媒体，也可用自身的 PT（人机界面），或语音单元，做到图、文、声并茂，或使用触摸

屏，进行操作，以方便使用。

这方面的发展虽刚起步，但发展势头较快，有的虽不是 PLC 的生产厂商，也为此在作努力。如人机界面，类似于 OMRON 的 PT，沈阳鹭岛公司也有自主产权的产品，我国的一些公司也作了专门开发。这些都可与多个厂家的 PLC 配套使用。

可见，使控制界面更加友好，对系统的操作更加方便，使控制的系统透明度更高，也是 PLC 发展的明显趋势。

便利化还应体现在编程上。用梯形图编程比助记符虽有很大进步，但还是不便。最好能面向对象编程，可视化编程、组态化编程。最好能，根据要求实现的功能，选选对象或组件，根据对象或组件间的关系连连线，根据实际使用的数据填填表格，就是编程，那多方便。

再就是便利化还应体现在软、硬件的设定上。硬件最好能实现在线插拔，即插即用。软件的设定，最好能随时设定，随时就起作用。

5. 进一步热备

PLC 工作非常可靠，但是，现实系统对控制可靠性的要求也在提高。因为随着生产效率的提高，系统的工作负荷将进一步加重，而且系统也越来越庞大与复杂。系统越复杂，如不采取措施，其产生故障的几率总是在增加的。所以，PLC 性能在提高的同时，也要采用措施，确保其工作可靠。

一个关键系统，一旦控制失灵，损失将是很大的。为此，重要的 PLC 控制系统或环节要求热备或容错，以做到万无一失，这是重要的。也是 PLC 应用的一个趋势。

特别是对那些连续生产系统，停工损失是很可观的。有时停工一天，其损失比热备的投入要多得多。所以，这些部门多将不惜重金，作冗余配置。

看看 DCS 产品，几乎所有的模块及联接的网络都是冗余的，难道 PLC 厂家就坐而视之，不想使 PLC 也在冗余模块生产与配置上再前进一步。

事实上，PLC 这方面的进步也是明显的。以 OMRON PLC 为例，过去只是 CPU 模块冗余，现在电源模块也可冗余，有的网络模块也可冗余。

其它厂商的冗余产品也不少。相信，将有更多的这类产品与配置问世。

6. 进一步趋同

微电子技术、计算机技术及通信技术的发展，为 PLC 的发展提供了基础。而 PLC 相关技术的发展又为 PLC 发展提供了借鉴。

反过来，PLC 的发展，对微电子技术、计算机技术及通信技术的发展，也是个促进。对 IPC、NC、DCS 的发展也将有所启示。

这些技术谁取代谁，大概是不可能的。因为谁都是基于计算机、通信及微电子技术，都可以完善自身，都可适应新的要求，谁也不是一成不变的。

总的趋势是，大家都在吸收对方的优点，发扬自身的特长，而且所基于的技术又是共同的，加上系统越来越开放，其结果将越来越趋向同一，很可能在一个现实系统中，这几者都用，或各有其特定的位置。

这 6 条是作者 1998 年在《可编程控制器配制 编程 联网》一书中提出的。如今时过 10 年，PLC 的向何处去，恐怕还是如此。在这几个“进一步”之后的 PLC 将是怎样的呢？

PLC 的概念还将更新，PLC 的应用还将扩展，PLC 的类型还将增多，PLC 不仅还将是工

业控制的一个重要支柱，而且，信息处理系统的一个重要支柱。而且，PLC 将不再是一个个体，而是一个智能化的工作控制装置及数据处理装置的大家族，单一产品的 PLC 将变成 PXC 家族。在这个大家族中，将有结构各一、性能各一、用途各一的，既有很大、又有很小的，既有简单、又有复杂的 PXC 产品。有 PLC、PPC、PMC、PRC、PCC、R（软）PLC。还有 GE 新推出的 PAC 等等。

图 10-8 所示为 GE PAC、RX7i 的外观。是 GE Fanuc 2003 年推出的第一个 PACSystems 平台。它基于标准嵌入式体系结构，比现有 PLC 的速度快 4 倍，为程序编制和文档存储提供最大为 10MB 的存储器。可通过以太网、Profibus、DeviceNet 以及 Genius 网络支持分布式 I/O。GE Fanuc 的 Proficy Machine Edition 软件为编程、配置和诊断提供工程开发环境。



图 10-8　PAC 外观

其实，PAC（Programmable Automation Controller）是可编程自动控制器简称。是 ARC Advisory Group 新造的一个词。PAC 的概念定义为：控制引擎的集中，涵盖 PLC 用户的多种需要，以及制造业厂商对信息的需求。PAC 包括 PLC 的主要功能和扩大的控制能力，以及 PC-based 控制中基于对象的、开放数据格式和网络能力等功能。

从图 10-8 知，只从外部看，PAC 和 PLC 区别不大，但在内部，有不少区别。它结合了 PLC 与 PC 中最好的特点，实现了 PC-based 的灵活、多功能性和高速度与 PLC 的耐用、高可靠性的完美结合。已成为 PLC 家族中最年轻，而又性能较高的一员。

第1版后记

《PLC 编程理论、算法及技巧》终于展现在您的面前了。为此，我首先要感谢她的第一位读者，我的“场外指导”，机械工业出版社电工电子分社牛新国社长及罗莉编辑。不是他们的策划及连续几个月的辛勤耕耘，我这本仅是一张光盘的书稿，怎么可能成为这么精致、工整、令人喜欢的正式出版物呢！

上海欧姆龙公司周珑先生、宋彤镖先生、董燕霞女士热心地为我的出书创造并提供了良好的环境与条件，给了我极其有力的支持与帮助。在此，我也衷心地对他们表示感谢！

本书作者标明的是我一人。但在实际上，我所做的只是文稿整理。真正的作者应该是鹭岛公司工控部、工程部及研发部的全体工程师们。

首先是周建先生，他现为鹭岛公司副总经理，主抓研发。在 PLC 应用及系统集成上作了很多实际工作。本书不少实例都与他的实际工作密切相关。

再就是鹭岛郑国刚副总经理、工控部唐克鲜副部长、工程部蔡永亮副部长、兰东辉工程师、尉红敏工程师、郑笑飞工程师、李天辉工程师、蒋辉工程师、薛勇工程师以及田维娜、冯喜、王洪博等同志，对我的书稿完成与定稿都给了很大的帮助。

最后要提到的是，鹭岛公司总经理张波先生。他非常精明，又富有魄力，是一位年轻有为的民营企业家。鹭岛今天的成功与他的有力领导及精心经营是分不开的。他对我的每次出书，都鼎力支持，多方帮助，此次出书更是如此。

所以，我要感谢张波先生，感谢周建先生，感谢所有帮助我的鹭岛公司的同志们。

还要特别提及的是，沈阳旭风电子科技开发有限公司总经理陈东旭高级工程师。他在可编程序控制器实验教学上进行了多年研究，设计了多种型号 PLC 的开关量控制、模拟量控制、脉冲量控制以及人机界面、传动系统的实验台与实验版。并把它产品化，成了沈阳旭风电子技术开发公司批量生产的，高等学校的教学仪器设备组成部分。他的这套实验装置已在东北大学等几十所高校成功使用。

为了帮助我出书，他不仅向我提出很多很好的创意与具体建议，还向我提供该公司生产的全套设备，为我进行书中有关程序测试提供了极大方便。在此，我要特别向他表示深深的谢意。

在我的书稿最后定稿中，史春卿、姜海涛、宋巍、谷秀范、张殿相、宋斌、洪颖、程希琳、田中心等也付出辛勤劳动，在此，也顺便向他们致以谢意！

最后应指出的是，本书成稿还得益于我有幸处于这个信息时代。正是这个时代出现的互联网技术，我才能非常方便、非常大量、非常准确、非常及时地收集到有关我所需要的写作参考资料。在此，我也向被我引用过资料的公司或作者，特别是 OMRON 公司，表示最诚挚的谢意！

最后要说的是，本书篇幅较大，图表很多，尽管我们尽心去检查、校对，但恐难免仍有些“笔误”，恳请读者不吝指正。

宋伯生

2004 年 8 月于沈阳

参 考 文 献

［1］ 机械工程手册电机工程手册编辑委员会．电机工程手册：第 9 卷．北京：机械工业出版社，1982.
［2］ 钱学森，宋健．工程控制论：上、下册．北京：科学出版社，1980.
［3］ OMRON 公司提供．有关 PLC 资料.
［4］ 宋伯生．机床电器及电控制器．北京：中国劳动出版社，1990.
［5］ 宋伯生．可编程控制器．北京：中国劳动出版社，1993.
［6］ 宋伯生．可编程控制器配置　编程　联网．北京：中国劳动出版社，1998.
［7］ 宋伯生．PLC 编程理论、算法及技巧．北京：机械工业出版社，2005.
［8］ 宋伯生．PLC 编程实用指南．北京：机械工业出版社，2007.
［9］ 宋伯生，陈东旭．PLC 应用及实验．北京：机械工业出版社，2006.
［10］ 宋伯生．PLC 系统配置及软件编程．北京：中国电力出版社，2007.
［11］ 和利时公司．有关 PLC 资料.

ISBN 978-7-111-26319-7
ISBN 978-7-89482-999-3(光盘)

在中国科技金书网上，本书经该网图书鉴定委员会鉴定，先评定为“经典图书”，后又评定为“优秀教材”。同时也获得网友的广泛赞许。

中国科技金书网
书评摘录

半月前我定购了这本书，它的确是本好书，内容深入浅出，作者实践经验丰富，读者对其内容能较快领会。较一些实用性的书却写得深奥晦涩，真是好得太多了，我已经看了大半本，受益很多，而且能很容易地用在工作中。需要多些这样的科技书。

本书从PLC原理、编程等方面详细介绍了PLC的工作过程。并以一个编程程序为例进行介绍。本书最精彩的部分是介绍了与计算机间的通信方法、通信程序的设计。是一本较好的书。这是其它此类书所没有的。其它PLC的书一般均为基本介绍。建议想提高PLC的朋友阅读。

这本书我买了。的确很不错，从基础到深入，从理论到实践，从开关量到模拟量，再到脉冲量等等，样样都有，并且讲得也很详细。建议学习工控技术的朋友多看看这本书，对大家PLC技术的提高会有很大帮助。

上架指导：工业技术/电气工程/PLC

地址：北京市百万庄大街22号
电话服务
社服务中心：(010)88361066
销售一部：(010)68326294
销售二部：(010)88379649
读者服务部：(010)68993821

邮政编码：100037
网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com
封面无防伪标均为盗版

定价：88.00元(含1CD)

